



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

“DESARROLLO DE UNA BIBLIOTECA PARA EL
MANEJO DE INTERFACES HUMANO-COMPUTADORA
A TRAVÉS DE GESTOS FACIALES”.

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN SISTEMAS
COMPUTACIONALES

PRESENTA:

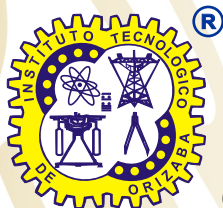
I.S.C. Fernando Octavio Reynoso Martínez

DIRECTOR DE TESIS:

M.C. María Antonieta Abud Figueroa

CODIRECTOR DE TESIS:

M.C. Silvestre Gustavo Sergio Peláez Camarena



ORIZABA, VERACRUZ, MÉXICO.

MARZO 2020



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

FECHA: 12/03/2020
DEPENDENCIA: POSGRADO
ASUNTO: Autorización de Impresión
OPCIÓN: I

C. **FERNANDO OCTAVIO REYNOSO MARTINEZ**
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

"DESARROLLO DE UNA BIBLIOTECA PARA EL MANEJO DE INTERFACES HUMANO-COMPUTADORA A TRAVES DE GESTOS FACIALES".

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E

MARIO LEONCIO ARRIJOA RODRIGUEZ
JEFE DE LA DIV. DE ESTUDIOS DE POSGRADO



Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz, México

Tel. 01 (272) 7 24 40 96, Fax. 01 (272) 7 25 17 28 e-mail: orizaba@itorizaba.edu.mx

www.orizaba.tecnm.mx





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

FECHA : 17/02/2020

ASUNTO: Revisión de Trabajo Escrito

C. MARIO LEONCIO ARRIJOJA RODRIGUEZ
JEFE DE LA DIVISION DE ESTUDIOS
DE POSGRADO E INVESTIGACION.
P R E S E N T E

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C. :

FERNANDO OCTAVIO REYNOSO MARTINEZ

la cual lleva el título de:

"DESARROLLO DE UNA BIBLIOTECA PARA EL MANEJO DE INTERFACES HUMANO-COMPUTADORA A TRAVES DE GESTOS FACIALES".

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE : M.C. MARIA ANTONIETA ABUD FIGUEROA

SECRETARIO : M.C. SILVESTRE GUSTAVO SERGIO PELAEZ

VOCAL : DRA. LISBETH RODRIGUEZ MAZAHUA

VOCAL SUP. : DR. ULISES JUAREZ MARTINEZ

ma. Antonieta Abud Figueroa
FIRMA

Silvestre Gustavo Sergio Pelaez
FIRMA

Lisbeth Rodriguez Mazahua
FIRMA

Ulises Juarez Martinez
FIRMA

EGRESADO (A) DE LA MAESTRIA EN SISTEMAS COMPUTACIONALES

OPCION: I Tesis



Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz, México

Tel. 01 (272) 7 24 40 96, Fax. 01 (272) 7 25 17 28 e-mail: orizaba@itorizaba.edu.mx

www.orizaba.tecnm.mx



Índice

Índice.....	i
Índice de Tablas.....	iii
Índice de Figuras.....	iv
Resumen.....	vi
Abstract.....	vii
Introducción.....	viii
CAPITULO 1. ANTECEDENTES	1
1.1. Marco teórico	1
1.1.2. Estilos de interactividad	1
1.1.2.5. Interfaz WIMP (Windows, iconos, menús, puntero)	3
1.1.3. Interacción Natural de Usuario (NUI)	5
1.1.4. Lenguaje verbal	5
1.1.5. Lenguaje No Verbal	6
1.1.6. Gestos faciales	6
1.1.7. Trayectoria del ojo humano	6
1.1.8. Dispositivos para Interacción Natural.....	7
1.1.9. Emotiv Epoc +.....	8
1.1.10. Java	8
1.1.11. Scrum	9
1.2. Planteamiento del problema.....	11
1.3. Objetivos general y específicos.....	12
1.3.1. Objetivo General.....	12
1.3.2. Objetivos Específicos.....	12
1.4. Justificación.....	12
Capítulo 2. Estado de la práctica.....	13
2.1. Trabajos relacionados.....	13
2.2 Análisis comparativo	18
2.3. Propuesta de la solución.....	23

2.3.1. Justificación de la tecnología seleccionada	23
Capítulo 3. Aplicación de la metodología	25
3.1. Descripción de la solución	25
3.1.1. <i>Sprint 1</i> Analizar dispositivo y su <i>suite</i> de desarrollo para la recepción de señales EEG.....	26
3.1.2. <i>Sprint 2</i> Definir los comandos que se incorporarán a la biblioteca que servirán para desarrollar interfaces gestuales.	29
3.1.3. <i>Sprint 3</i> . Diseñar la arquitectura de la biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.....	33
3.1.4. <i>Sprint 4</i> . Diseñar la arquitectura de la biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.....	35
3.1.5. <i>Sprint 5</i> . Incorporación de la biblioteca en el ambiente Java	54
Capítulo 4. Resultados	60
4.1. <i>Sprint 6</i> . Aplicación a un caso de estudio	60
4.1.1. Planteamiento del caso de estudio	60
4.1.2. Introducción	61
4.1.3. Objetivo general.....	63
4.1.4. Población atendida	63
4.1.5. Metodología	63
4.1.6. Implementación de la biblioteca en caso de estudio.....	64
4.3 Características de la implementación del caso de estudio.....	77
4.4 Resultado de Implementación de biblioteca para el manejo de interfaces humano-computadora a través de gestos faciales en un Sistema de modificación de conducta con niños de déficit de atención con o sin hiperactividad	78
Capítulo 5. Conclusiones y Recomendaciones	87
5.1. Conclusiones.....	87
5.2. Recomendaciones	89
Productos Académicos.....	90
REFERENCIAS.....	91

Índice de Tablas

Tabla 2. 1. Análisis comparativo de los trabajos relacionados.	18
Tabla 3. 1. Pila del producto.....	25
Tabla 3. 2..Sprint número uno.	26
Tabla 3. 3. Clases de SDK de Emotiv y su funcionamiento.	28
Tabla 3. 4. Sprint número dos.	29
Tabla 3. 5. Gestos faciales definidos para la biblioteca de interfaces gestuales... 31	
Tabla 3. 6. Gestos faciales descartados para el desarrollo de la biblioteca de interfaces gestuales.	32
Tabla 3. 7. Definición de los movimientos que actuarán dentro de la biblioteca de interfaces gestuales.	32
Tabla 3. 8. Sprint número tres.	33
Tabla 3. 9. Sprint número cuatro.	35
Tabla 3. 10. Sprint número Cinco.	54
Tabla 4. 1. Sprint número Seis.	60
Tabla 4. 2. Resultado de cuestionario Previo.	79
Tabla 4. 3. Gestos Faciales implementados en caso de estudio.....	80
Tabla 4. 3. Gestos Faciales implementados en caso de estudio.....	81
Tabla 4. 4. Comandos implementados en el caso de estudio.	81
Tabla 4. 4. Comandos implementados en el caso de estudio.	82
Tabla 4. 5. Comandos implementados en el caso de estudio.	82
Tabla 4. 6. Resultado de cuestionario posterior.	85
Tabla 4. 7. Resultado de cuestionario Posterior del Profesor.....	86

Índice de Figuras

Figura 3. 1. Diadema Emotiv Epoc + con electrodos colocados.	27
Figura 3 2.Página de descarga de EmotivBCI.....	27
Figura 3 3.Apartado de entrenamiento de expresiones faciales en EmoticBCI.....	28
Figura 3 4. Prueba de gestos faciales Guiño Derecho.	30
Figura 3 5. Aplicación muestra para verificación de funcionamiento de gestos faciales.	30
Figura 3 6. Arquitectura Patrón de arquitectura Modelo Vista Controlador.	34
Figura 3 7. Patrón de diseño Adapter.....	35
Figura 3 8. Clase Conexion.	36
Figura 3 9. Código de la Clase Apuntador.	37
Figura 3 10. Código del Método <code>clic()</code>	39
Figura 3 11. Contenido Paquete Principal.	40
Figura 3 12. Código Interfaz GestosFaciales.	41
Figura 3 13. Representación del Patrón de diseño Adapter para la obtención de gestos faciales.....	41
Figura 3 14. Clase <code>ParpadeoIzquierdo</code>	42
Figura 3 15. Clase <code>ParpadeoDerecho</code>	42
Figura 3 16. Clase <code>MiradaIzquierda</code>	43
Figura 3 17. Clase <code>MiradaDerecha</code>	44
Figura 3 18. Mirada hacia la izquierda.	44
Figura 3 19. Mirada hacia la derecha.	45
Figura 3 20. Parpadeo izquierdo.	45
Figura 3 21. Parpadeo derecho.....	46
Figura 3 22. Clase <code>Parpadeo</code>	47
Figura 3 23. Parpadeo.....	47
Figura 3 24. Clase <code>Sonrisa</code>	48
Figura 3 25. Clase <code>Fruncido</code>	48
Figura 3 26. Gesto Facial Fruncido.	49
Figura 3 27. Clase <code>Señales</code> y creación de Objetos.....	50
Figura 3 28. Variables y Objetos para el manejo de la biblioteca.....	50
Figura 3 29. Firma del método resultado, creación y asignación de variables en el método.	51
Figura 3 30. Implementación de los objetos.....	51
Figura 3 31. Manejo de eventos de Gestos Faciales.	52
Figura 3 32. Selección de funcionalidades de los Gestos Faciales.....	53
Figura 3 33. Método <code>mouseDefault()</code>	54
Figura 3 34. Archivo <code>BIG.jar</code>	55
Figura 3 35. Importar archivo <code>BIG.jar</code>	55
Figura 3 36. Selección de archivo <code>BIG.jar</code>	56
Figura 3 37. Archivo <code>BIG.jar</code> cargado en el proyecto.....	56

Figura 3 38. Importación de Clase Señales de la Biblioteca de Interfaces Gestuales.	58
Figura 3 39. Importar Clases de la Biblioteca de Interfaces Gestuales.	58
Figura 3 40. Implementación de la Biblioteca de Interfaces Gestuales.	59
Figura 4. 1. Diagrama de Clases de la implementación de la biblioteca de interfaces gestuales en caso de estudio.	64
Figura 4. 2. Pantalla de Inicio.	65
Figura 4. 3. Pantalla de Inicio.	65
Figura 4. 4. Tablero donde cuyo objetivo es identificar la cabeza.	66
Figura 4. 5. Error en el tablero por no haber seleccionado la imagen de la cabeza.	66
Figura 4. 6. Error en el tablero por no haber seleccionado la figura del brazo.	67
Figura 4. 7. Ventana cuando se selecciona la figura brazo correctamente.	67
Figura 4. 8. Error en el tablero por no haber seleccionado la figura de los pies.	68
Figura 4. 9. Ventana al seleccionar la figura pies correctamente.	68
Figura 4. 10. Error en el tablero al haber seleccionado el lápiz pequeño.	69
Figura 4. 11. Ventana al seleccionar el lápiz más grande.	69
Figura 4. 12. Error en el tablero al haber seleccionado imagen incorrecta.	70
Figura 4. 13. Ventana que aparece al haber elegido la figura correcta.	70
Figura 4. 14. Error en el tablero al haber elegido imagen con menos lápices.	71
Figura 4. 15. Ventana al haber seleccionado la figura que contiene más lápices.	71
Figura 4. 16. Error en el tablero al haber elegido figura donde es de noche.	72
Figura 4. 17. Ventana al haber elegido la figura que contiene más lápices.	72
Figura 4. 18. Error en el tablero al haber elegido figura donde es de noche.	73
Figura 4. 19. Ventana al haber elegido la figura correcta donde el niño duerme.	73
Figura 4. 20. Actividad para identificar el color verde.	74
Figura 4. 21. Actividad para identificar el color amarillo.	74
Figura 4. 22. Actividad de identificar un cilindro.	75
Figura 4. 23. Actividad de identificar una cruz.	75
Figura 4. 24. Actividad de identificar una cara al revés.	76
Figura 4. 25. Actividad de identificar una luna al revés.	76
Figura 4. 26. Dispositivos para el manejo de biblioteca de interfaces gestuales.	78
Figura 4. 27. Entrenamiento Gestual.	80
Figura 4. 28. Participación en caso de estudio.	84

Resumen

Las interfaces humano-computadora se basan principalmente en el modelo de ventanas, punteros y ratones como paradigma dominante para la interacción con las aplicaciones. El uso de gestos es una adición valiosa al repertorio de técnicas de interacción humano-computadora, sin embargo, requieren tiempo para desarrollarse mejor. Como toda nueva tecnología presenta nuevos desafíos, entre los cuales se encuentra el contar con herramientas que soporten el desarrollo de aplicaciones que incluyan este tipo de interacción. Esto conlleva a la necesidad de desarrollar la infraestructura de soporte para el manejo de interfaces a través de gestos.

Los objetivos del trabajo de tesis fueron desarrollar una biblioteca para Java Swing, mediante las tecnologías Java, Emotiv Epoc +, Emotiv SDK, aplicando la metodología SCRUM, para el manejo de interfaces humano-computadora a través de gestos faciales para posteriormente llevar a cabo un caso de estudio donde se implementó la biblioteca. De esta manera, los programadores contarán con más herramientas que le permitan incluir interfaces gestuales en sus desarrollos.

Abstract

Human-computer interfaces are mainly based in the window, pointer, mouse model as a dominant paradigm for the interaction with applications. The use of gestures is a valuable addition to the repertoire of human-computer interaction techniques, however, they require time to develop better. Like every new technology it presents new challenges, among them it is to have tools that support the development of applications that include this type of interaction. This leads to the need of developing the support infrastructure for the managing of interfaces through gestures. The objectives of the thesis work were to develop a library for Java Swing, through Java technologies, Emotiv Epoc +, Emotiv SDK, applying the SCRUM methodology, for the management of human-computer interfaces through facial gestures to later lead to a case study where the library was implemented. In this way, programmers will have more tools that allow them to include gestural interfaces in their developments.

Introducción

Las *Interfaces humano computadora* investigan y tratan todos los aspectos relacionados con el diseño y la implementación de las interfaces entre los humanos y las computadoras. Las *Interfaces humano computadora* involucran múltiples disciplinas relacionadas con la ciencia de la computación, (procesamiento de imágenes, visión computarizada, lenguajes de programación y otras similares), así como disciplinas relacionadas con las ciencias humanas (ergonomía, factores humanos, psicología cognitiva, y otras similares).

Las interfaces humano-computadora actualmente son esenciales para facilitar la vida de muchas personas, ya sea para la ayuda en la vida diaria, entretenimiento o para mejorar la calidad de vida.

El uso de gestos es una adición valiosa al repertorio de técnicas de interacción humano-computadora, los sistemas gestuales son uno de los caminos futuros importantes para una interacción más holística y humana de las personas con la tecnología, representando un gran potencial para mejorar la interacción. Los avances tecnológicos de las últimas décadas proporcionan nuevos mecanismos de interacción.

El presente trabajo de investigación se encuentra estructurado en cinco capítulos: el capítulo uno son los Antecedentes, se presentan los conceptos relacionados con el tema de investigación. De igual forma se incluye el problema a resolver, los objetivos y la justificación del trabajo. En el capítulo dos se encuentra el Estado de la Práctica donde se obtuvieron y revisaron meticulosamente diversos artículos relacionados con el mismo trabajo de tesis, de los cuales se extrajo la información más sobresaliente. El capítulo tres aplicaciones de la metodología donde se da a conocer el desarrollo de la solución. En el capítulo cuatro resultados se muestran los resultados de la implementación de la biblioteca de interfaces gestuales. El capítulo cinco conclusiones y recomendaciones se presentan las conclusiones al finalizar el proyecto tesis y con ello se dan recomendaciones para trabajar en un futuro.

CAPITULO 1. ANTECEDENTES

En este capítulo se presentan los conceptos relacionados con el tema de investigación. De igual forma se incluye el problema a resolver, los objetivos y la justificación del trabajo.

1.1. Marco teórico

A continuación, se definen algunos términos relevantes para el tema de investigación.

1.1.1. Interfaz Humano-Computadora

HCI (Human Computer Interfaces, Interfaz humano-computadora) investiga y trata todos los aspectos relacionados con el diseño y la implementación de las interfaces entre los humanos y las computadoras. Dada la naturaleza y objetivos, la HCI en forma innata involucra múltiples disciplinas relacionadas con la ciencia de la computación, (procesamiento de imágenes, visión computarizada, lenguajes de programación y otras similares), así como disciplinas relacionadas con las ciencias humanas (ergonomía, factores humanos, psicología cognitiva, y otras similares). La investigación sobre HCI primariamente concierne al diseño, implementación y especificación de nuevas interfaces para mejorar la interacción entre humanos y máquinas. El término mejorar se relaciona con diferentes aspectos, incluyendo lo intuitivo en el uso, así como la robustez de una interfaz.

Una interfaz natural, intuitiva, eficiente, robusta y configurable reduce fuertemente el espacio entre los modelos mentales humanos y la forma en que las computadoras, máquinas o robots, desarrollan sus tareas. Sin embargo, estudios sobre HCI, que se remontan a 1975 y avances tecnológicos recientes en electrónica de consumo abrieron nuevos escenarios excitantes: gestos, posturas corporales de manos, vocalizaciones y miradas son solo algunos de los nuevos modos de interacción que se utilizan para diseñar interfaces de usuario naturales (NUIs) [1].

1.1.2. Estilos de interactividad

La interacción se ve como un diálogo entre la computadora y el usuario. Algunas aplicaciones tienen estilos de interacción muy distintos [2].

Se identifican algunos estilos comunes:

- Interfaz de línea de comando
- Menús
- Lenguaje natural
- Diálogo pregunta / respuesta y consulta
- Formularios y hojas de cálculo
- WIMP (*Windows, Icons, Menu, Pointers-ventanas, iconos, menús y puntero*).

1.1.2.1. Interfaz de línea de comando

Forma de expresar instrucciones a la computadora directamente. Son teclas de función, caracteres individuales, abreviaturas cortas, palabras completas o una combinación [3]:

- Adecuado para tareas repetitivas.
- Mejor para usuarios expertos que principiantes.
- Ofrecen acceso directo a la funcionalidad del sistema.
- Los nombres / abreviaturas de los comandos deberían ser significativos.

1.1.2.2. Menús

Conjunto de opciones que se muestran en la pantalla. Las opciones son visibles, por lo que exigen menos recordatorios: se confía en el reconocimiento para que los nombres sean significativos. Seleccionado usando las teclas del *mouse*, numéricas o alfabéticas. A menudo las opciones se agrupan jerárquicamente: se necesita una agrupación razonable.

Los sistemas de menú se clasifican en:

- Basados en texto, con opciones presentadas como opciones numeradas.
- Con un componente gráfico, con el menú que aparece en el recuadro y las elecciones hechas ya sea escribiendo la letra inicial o moviéndose con las teclas de flecha [3].

1.1.2.3. Lenguaje natural

Una opción atractiva: usan reconocimiento de voz común o lenguaje natural escrito [3].

1.1.2.4. Interfaces de llenado de formularios

- Principalmente para entrada de datos o recuperación de datos.
- Pantalla como formulario en papel.
- Los datos se colocan en un lugar relevante.
- Requiere un buen diseño e instalaciones de corrección obvias [3].

1.1.2.5. Interfaz WIMP (Windows, iconos, menús, puntero)

WIMP es un acrónimo que surgió en la década de 1980 y describe la interfaz gráfica de usuario (GUI) de las computadoras personales. Incluye las interfaces de Windows y Macintosh, así como otros sistemas operativos menos comunes, como Linux y NeXT.

Si bien los términos GUI y WIMP a veces se usan indistintamente, WIMP es técnicamente un subconjunto de GUI. Esto significa que todas las interfaces WIMP son GUI, pero no todas las GUI son WIMP. Los sistemas basados en WIMP están diseñados para usarse con un teclado y un ratón, ya que el *mouse* controla el puntero (o cursor) y el teclado se usa para ingresar datos. Otras GUI pueden admitir diferentes tipos de entrada, como una pantalla táctil [3].

1.1.2.5.1. Windows

Áreas de la pantalla que se comportan como si fueran terminales independientes

- Contiene texto o gráficos.
- Se mueve o redimensiona.
- Se superpone y oscurecerse entre sí, o se coloca uno al lado del otro.

- Las barras de desplazamiento permiten al usuario mover los contenidos de la ventana hacia arriba y hacia abajo o de lado a lado.
- Las barras de título describen el nombre de la ventana [3].

1.1.2.5.2. Iconos

Imagen pequeña o imagen, utilizada para representar algún objeto en la interfaz, a menudo una ventana. Se permite que la ventana (*window*) se cierre a esta pequeña representación (iconizada) permitiendo que muchas ventanas sean accesibles. Los iconos son muchos y diversos: representaciones muy estilizadas o realistas.

1.1.2.5.3. Punteros

En el ámbito de las *HCI* el puntero se refiere al símbolo que representa los movimientos del *mouse*.

Componente importante, ya que el estilo WIMP se basa en señalar y seleccionar cosas como iconos y elementos de menú.

- Generalmente se logra con el *mouse*.
- *Joystick*, *Trackball*, teclas de cursor o atajos de teclado también se utilizan.
- Amplia variedad [3].

1.1.2.5.4. Menús

Elección de las operaciones o servicios a realizar ofrecidos en la pantalla.

Opción requerida seleccionada con puntero.

- Problema: los menús algunas veces ocupan mucho espacio en la pantalla.
- Solución: menús desplegados o menús emergentes.
- Los menús desplegados se arrastran hacia abajo desde un solo título en la parte superior de la pantalla.
- Los menús emergentes aparecen cuando se hace clic en una región particular de la pantalla.

Algunos menús son *pin-up* permanecen en la pantalla hasta que se solicita explícitamente que se vayan. Otro tipo es el menú de bajada, similar al menú desplegable, pero no es necesario seleccionar explícitamente la barra.

- También menús en cascada: una selección de menú abre otro menú adyacente, y así sucesivamente.
- Menú *Pie*: opciones de menú organizadas en círculo.

Más fácil de seleccionar elemento (área de destino más grande) y a veces se ofrecen aceleradores de teclado más rápidos - combinaciones de teclas que tienen el mismo efecto que seleccionar el elemento del menú [3].

1.1.3. Interacción Natural de Usuario (NUI)

Una interfaz natural de usuario (NUI) es un sistema para la interacción humano-computadora que el usuario opera a través de acciones intuitivas relacionadas con el comportamiento humano natural y cotidiano.

Una NUI se opera de diferentes maneras, dependiendo del propósito y los requisitos del usuario. Algunas NUI dependen de dispositivos intermediarios para la interacción, pero las NUI más avanzadas son invisibles para el usuario o tan discretas que parecen invisibles rápidamente [4].

La ventaja de las NUI es que el usuario percibe la interacción divertida, fácil y natural porque el usuario utiliza una gama más amplia de habilidades básicas en comparación con la interacción de interfaz gráfica de usuario más tradicional, que ocurre principalmente a través de un *mouse* y un teclado. Una NUI imita la interacción del usuario con el mundo físico al tener una correlación directa entre la acción del usuario y la reacción de NUI [5].

1.1.4. Lenguaje verbal

La comunicación verbal representa aquella en la que se usan las palabras, los signos sonoros o los auditivos; es el uso de las palabras para la interacción de manera invariable a la forma en la que las utilicen dependiendo del contexto. Se realiza de dos formas principales: oral (con el uso de palabras) o escritas (por medio de representación gráfica de signos). La comunicación oral goza de un amplio rango de formas: gritos, exclamaciones, silbidos, risas, lloros, sonidos vocales, entre otros. Todos ellos se expresan en multitud de ocasiones en sustitución del lenguaje formal y articulado (idioma) que es la forma de comunicación oral más desarrollada y clara para entender. El lenguaje articulado son sonidos estructurados que dan

lugar a las sílabas, palabras y oraciones con las que se da la comunicación con las demás personas [6].

1.1.5. Lenguaje No Verbal

La influencia de la conducta no verbal en la interacción humana es indiscutible. El dominio del lenguaje corporal resulta de especial utilidad en algunas funciones de la socialización.

La conducta no verbal se expresa principalmente a través de ocho canales, esos canales son:

- 1) Expresiones faciales.
- 2) Gestos.
- 3) Posturas.
- 4) Apariencia.
- 5) Háptica.
- 6) Proxémica.
- 7) Lenguaje corporal.
- 8) Paralenguaje [7].

1.1.6. Gestos faciales

Los gestos faciales son todos aquellos gestos provenientes de la cara. Dichos gestos expresan mucho sobre el estado de una persona o cómo piensa. La expresión facial permite la comunicación, adecuación y regulación de las emociones dentro del contexto social. En este sentido, la capacidad de discriminar las emociones a través de la expresión facial favorece la seguridad y eficacia de diferentes profesionales [8].

1.1.7. Trayectoria del ojo humano

El seguimiento ocular, o interacción a través de la mirada, es una tecnología que se usa para saber adónde mira una persona que está viendo la pantalla de una computadora. La tecnología también se utiliza para controlar una computadora con los ojos en lugar de usar el teclado y *mouse* tradicionales.

El seguimiento ocular, o interacción a través de la mirada, no es un concepto nuevo. Sin embargo, recientemente surgió como una solución viable para ayudar a que las

personas con discapacidad cognitiva y física tengan vidas más plenas e independientes [2].

1.1.8. Dispositivos para Interacción Natural

Diferentes tareas, diferentes tipos de datos y diferentes tipos de usuarios requieren diferentes dispositivos de interfaz de usuario. En la mayoría de los casos, los dispositivos de interfaz son dispositivos de entrada o dispositivos de salida, aunque, por ejemplo, una pantalla táctil combina ambos. En cualquier caso, los dispositivos disponibles proporcionan el conjunto de marcos para la configuración de interacción. Los dispositivos de interfaz se correlacionan con los sentidos humanos. Hoy en día, un dispositivo generalmente está diseñado para entrada o para salida [3].

1.1.8.1. Los dispositivos de entrada

Lo más común es que las computadoras personales estén equipadas con entrada de texto y dispositivos señaladores. Para la entrada de texto, el teclado QWERTY es la solución estándar, pero dependiendo de la finalidad del sistema, los dispositivos de entrada más especializados como teclados especiales, escáner con reconocimiento de caracteres, pluma o incluso entrada de voz pueden ser la mejor opción.

Al mismo tiempo, el *mouse* no es el único dispositivo señalador imaginable. Las alternativas para fines similares, pero ligeramente diferentes incluyen *touch pad*, *trackball*, *joystick* o incluso la mirada, al igual que los dispositivos para manipulación 3D.

Hay que tener en cuenta que la manipulación 3D es una cuestión de no solo moverse a una ubicación particular, sino también elegir una orientación particular. Para determinar el cabeceo, guiñada y balanceo, además de la ubicación, se requieren seis grados de libertad (valores independientes que representan las direcciones en las que puede ocurrir un movimiento), no solo tres [3], por ejemplo: moverse hacia delante/atrás, arriba/abajo, izquierda/derecha (traslación en tres ejes perpendiculares).

1.1.8.2. Dispositivos de salida

La salida de una computadora personal en la mayoría de los casos significa la salida de datos visuales.

Los dispositivos para la visualización "dinámica" incluyen el tubo de rayos catódicos (CRT) tradicional, la pantalla de cristal líquido (LCD) o dispositivos especializados como la pantalla de visualización de un piloto. Las impresoras también son un dispositivo muy importante para la producción visual, pero son sustancialmente diferentes de las pantallas en que su salida es estática: no cambiará con el tiempo, excepto por el color amarillento del papel.

Para aumentar el ancho de banda de la información que llega al usuario, es un objetivo importante usar más canales además de la salida visual. Un suplemento comúnmente utilizado para la información visual es el sonido, pero su verdadero potencial a menudo no se reconoce. La retroalimentación audible puede hacer que la interacción sea mucho más cómoda para el usuario, proporcionando información inequívoca sobre el estado del sistema y el éxito o fracaso de la interacción (por ejemplo, presionar un botón), sin poner aún más carga en el canal visual [3].

1.1.9. Emotiv EPOC +

Emotiv EPOC + es un electroencefalograma multicanal (sistema EEG) que está diseñado para la investigación del cerebro humano escalable y contextual y aplicaciones avanzadas de interfaz cerebro-computadora y brinda acceso a datos cerebrales de grado profesional, cuenta con 14 canales, métricas de rendimiento: emoción, compromiso, relajación, interés, estrés, enfoque, su método de muestreo es un muestreo secuencial, ADC (*Application Delivery Controller*, *Controladores de Entrega de Aplicaciones*) único. Sus sensores son almohadillas de fieltro y deben empaparse en solución salina antes de usarse. Es multiplataforma, permite trabajar con ella en Windows, Mac, Android y IOS. Su frecuencia de muestreo es de 32/64/128 Hz. Cuenta con reconocimiento de expresiones faciales (parpadeo, guiño, sorpresa, fruncir el ceño, sonreír, apretar) [9].

1.1.10. Java

Es un lenguaje de programación orientado a objetos, de alto nivel que al mismo tiempo es compilado e interpretado. Java se creó como una herramienta de programación para usarse en un proyecto de *set-top-box* en una pequeña operación

denominada *the Green Project* en Sun Microsystems en el año 1991. El compilador se encarga de convertir el código fuente de un programa en un código intermedio llamado *bytecode*, el cual es independiente de la plataforma en que se trabaje y se ejecuta por el intérprete de Java que forma parte de la Máquina Virtual de Java.

Java toma mucha de su sintaxis de C y C++, pero su modelo de objetos es más simple y elimina las herramientas de bajo nivel, por lo tanto, Java es un lenguaje de programación orientado a objetos de alto nivel, propósito general, concurrente, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible, por lo cual ofrece las siguientes características [10]:

- **Simple:** Java elimina muchas características de otros lenguajes, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el recolector de basura, el cual se encarga de liberar la memoria sin tener que hacerlo el programador, lo que permite liberar grandes bloques de memoria.
- **Orientado a objetos:** Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.
- **Distribuido:** Java realmente no es distribuido, pero proporciona las bibliotecas y herramientas necesarias para que los programas lo sean.
- **Robusto:** Java realiza comprobaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. Implementa los arreglos auténticos, en lugar de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir en memoria, resultado de punteros que señalan a zonas equivocadas.
- **Arquitectura natural:** El código fuente Java se compila a un código de bytes de alto nivel independiente de la máquina.
- **Seguro:** En Java, características como los punteros o el *casting* implícito que hace el compilador de C y C++ se eliminan para prevenir el acceso ilegal a la memoria.
- **Multihilo:** Java permite muchas actividades simultáneas en un programa.

1.1.11. Scrum

Es un marco para desarrollar, entregar, mantener y gestionar proyectos complejos, dentro del cual se abordan problemas complejos de adaptación. Trabaja de forma iterativa e incremental. Después de cada iteración se crea un entregable que da valor de negocio al cliente.

Scrum se adapta a cambios en los requerimientos del proyecto. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos

y regulares. De esta manera es posible adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente.

La gestión de los proyectos en Scrum se centra en definir las características de los proyectos a construir y quitar los obstáculos que puedan interferir las tareas del equipo de desarrollo.

El equipo Scrum está conformado por un maestro Scrum, un equipo de desarrollo y el propietario del producto. Los equipos seleccionan la mejor manera para desarrollar su trabajo sin depender de gente que no esté dentro del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

El equipo Scrum entrega productos de forma iterativa e incremental, maximizando las oportunidades de retroalimentación. Las entregas incrementales del producto "Hecho" aseguran que una versión potencialmente útil del producto de trabajo esté siempre disponible.

Scrum describe cuatro eventos formales para la inspección y adaptación del proyecto en el que se esté trabajando, los cuales son:

- Planificación de *Sprint*
- Scrum diario
- Revisión de *Sprint*
- *Sprint* retrospectiva

Scrum cuenta con tres artefactos predefinidos que representan trabajo o valor para brindar transparencia y oportunidades de inspección y adaptación con el proyecto que se trabaja. Los cuales son [11]:

- Pila de Producto
- *Sprint Backlog*
- Incremento

3.2.3.1 NetBeans

NetBeans es un IDE (*Integrated Development Environment*, entorno de desarrollo integrado) de licencia gratuita el cual no cuenta con restricciones de uso, NetBeans es el IDE oficial para Java 8 [12].

En NetBeans las aplicaciones se crean a partir de módulos, que permiten extenderse agregando nuevos módulos desarrollados independientemente, lo que hace que las aplicaciones hechas en NetBeans sean extendidas fácilmente por otros desarrolladores.

El editor de NetBeans recompone fácilmente el código, porque cuenta con una gran gama de herramientas, mientras que también proporciona plantillas de código, sugerencias de codificación y generadores de código. El editor soporta diversos lenguajes de programación como son: Java, C / C ++, XML (*Extensible Markup Language, Lenguaje de Mercado Extensible*) y HTML (*HyperText Markup Language, Lenguaje de Marcas de Hipertexto*), hasta PHP, Groovy, JavaScript y JSP (*Java Server Pages, Páginas de Servidor Java*). Debido a que el editor es extensible, ofrece soporte para muchos otros idiomas.

Mantiene una visión general clara de las aplicaciones grandes, con miles de carpetas y archivos, y millones de líneas de código. Diseña las interfaces gráficas de usuario para las aplicaciones Java SE (*Standard Edition, Edición Estandar*), HTML5, Java EE (*Enterprise Edition, Edición Empresarial*), PHP, C / C ++ y Java ME (*Micro Edition, Edición Micro*) de forma rápida y sin problemas mediante el uso de editores y herramientas.

NetBeans proporciona herramientas de análisis estático, fundamentalmente de integración con la herramienta FindBugs ampliamente utilizada, para identificar y solucionar problemas comunes en el código Java [12].

1.2. Planteamiento del problema

Las interfaces humano-computadora se basan principalmente en el modelo de ventanas, punteros y ratones como paradigma dominante para la interacción con las aplicaciones, sin embargo, los avances tecnológicos de las últimas décadas proporcionan nuevos mecanismos de interacción, entre los cuales se encuentran dispositivos capaces de detectar movimientos del cuerpo humano, lo que ha conducido a un incremento en el uso de interfaces basadas en gestos realizados con las manos o la cara, lo que se denomina Interfaces de Usuario Naturales (NUI, por sus iniciales en inglés).

La investigación y el desarrollo en interfaces naturales y gestuales es abundante siendo el potencial más interesante en este campo lograr interfaces gestuales sin cables. Las primeras soluciones planteadas requerían guantes caros u otro equipamiento intrusivo. En la medida que los sensores y microprocesadores se abaratan y se hacen más potentes, la posibilidad de construir interfaces capaces de recibir e interpretar los más variados gestos sin vinculación cableada se convierte en realidad. Los últimos trabajos de investigación en el campo se focalizan en interfaces de detección de movimiento libre y las expresiones faciales. Sin embargo, en la mayoría de los casos se trata de trabajos enfocados en un problema en particular, y no se encuentran reportes de herramientas que permitan a desarrolladores de software la fácil incorporación de estas interfaces en su trabajo.

1.3. Objetivos general y específicos

A continuación, se muestra el objetivo general y lo específicos.

1.3.1. Objetivo General

Desarrollar una biblioteca para aplicaciones de escritorio en Java Swing a través de gestos faciales.

1.3.2. Objetivos Específicos

- Revisar tecnologías de desarrollo de interfaces humano-computadora con control gestual para determinar la más adecuada para el proyecto.
- Seleccionar el dispositivo de adquisición y el software para análisis de las señales que se utilizarán en el desarrollo.
- Definir los comandos que se incorporarán a la biblioteca que servirán para desarrollar interfaces gestuales.
- Establecer una arquitectura que sirva de base al desarrollo de la biblioteca.
- Desarrollar la biblioteca para la plataforma de desarrollo elegida.
- Probar la biblioteca en un caso de estudio.

1.4. Justificación

El uso de gestos es una adición valiosa al repertorio de técnicas de interacción humano-computadora, sin embargo, requieren tiempo para desarrollarse mejor, para que los desarrolladores comprendan cómo implementarlos y para que se desarrollen las convenciones estándar que regulen su significado de modo que los mismos gestos signifiquen lo mismo en diferentes sistemas. Esto conlleva a la necesidad de desarrollar la infraestructura de soporte para el manejo de interfaces a través de gestos.

Los sistemas gestuales son uno de los caminos futuros importantes para una interacción más holística y humana de las personas con la tecnología, representando un gran potencial para mejorar la interacción.

Sin embargo, como toda nueva tecnología presenta nuevos desafíos, entre los cuales se encuentra el contar con herramientas que soporten el desarrollo de aplicaciones que incluyan este tipo de interacción. Por esta razón, en este trabajo se propone el desarrollo de una biblioteca que permita incluir interfaces gestuales en nuevos productos de software.

Capítulo 2. Estado de la práctica

Como una introducción al tema se obtuvieron y revisaron meticulosamente diversos artículos relacionados con el mismo, de los cuales se extrajo la información más sobresaliente. El resultado del análisis de los artículos revisados es que las interfaces humano-computadora y los gestos faciales son importantes hoy en día en el desarrollo de aplicaciones para personas con impedimentos físicos motrices. Por lo cual, cada artículo revisado hizo una aportación a esta investigación. En este capítulo se describen los artículos revisados y relacionados con las interfaces humano-computadora y los gestos faciales, posteriormente se muestra una tabla comparativa de cada uno de ellos y se concluye con los resultados obtenidos de todos estos.

2.1. Trabajos relacionados

Hawkes et al. [13] establecieron como propósito probar la confiabilidad y utilidad de la diadema Emotiv EPOC para determinar si se debe utilizar en el campo médico, ya que se piensa que una tecnología como ésta ayudará a aquellos que sufren de parálisis debidas a enfermedades tales como la ALS (Esclerosis lateral amiotrofia), ya que la enfermedad no tiene cura es importante una solución que ayude a mejorar las necesidades médicas. Para la solución se propuso contemplar el uso de la diadema Emotiv EPOC, para el control del robot Finch mediante expresiones faciales, se concluyó que la diadema Emotiv Epoc tuvo éxito al ser conectada y controlar al robot Finch, pero con algunas fallas el dispositivo reconoció los movimientos de las personas, así como también que la diadema no es médicamente funcional o confiable.

Hornero et al. [14] mencionaron la existencia de soluciones previas para ayuda de la discapacidad utilizando TIC'S, a pesar de contar con soluciones existentes sus costos son elevados, requieren de procesos de calibración por cada usuario, lo que limita su usabilidad, por lo que su propuesta usa un acelerómetro ADXL345, el microcontrolador ATmega328A, el driver FTDI232RL, una placa de circuito impreso, cinta elástica, Bluetooth (módulo RN41-XV), Labview para desarrollar un módulo de modo Ratón y modo Teclado. El modo Ratón asocia cada movimiento del usuario a las direcciones del ratón. El modo Teclado se presenta en dos opciones de funcionamiento. En la primera se asocian cuatro movimientos escogidos con anterioridad por el usuario y en la segunda opción, si el usuario presenta una mayor habilidad motora, se asocian los diferentes movimientos que el usuario haya escogido.

Vidal et al. [15] resaltaron la importancia de tomar en cuenta las características de los ojos cuando se busca detectar si se sigue un objeto en movimiento, sobre todo el hecho de que esta característica de la orientación de objetos en movimiento es más difícil explotar que la orientación de objetos estáticos, por lo que se necesita un método diferente a los anteriores. Su propuesta usó una pantalla grande (89x50 cm, 1920x1080px) y un seguidor ocular remoto Tobii X300. Se obtuvo como resultado final el método Pursuits y tres aplicaciones interactivas.

Liang et al. [16] afirmaron que es importante contar con una solución de tecnologías de seguimiento de manos y reconocimiento de gestos para simplificar las operaciones y proporcionar una interfaz intuitiva, en la que los niños utilicen gestos con la mano para manipular marionetas virtuales para realizar historias. Propusieron usar el controlador Leap Motion, el Leap Motion SDK (*software development kit, kit de desarrollo de software*) proporcionado por Leap Motion como dispositivo sensor *HCI (Human-Computer Interaction, Interacción Humano-Computadora)* para rastrear gestos con las manos, Leap Motion SDK como *API (Application Programming Interface, Interfaz de Programación de Aplicaciones)* para acceder a los datos de movimiento de las manos y los dedos del dispositivo, Maya 2014 para crear todos los modelos y animaciones 3D y Unity3D Pro V4.2 para integrar y desarrollar todo el sistema. La arquitectura del sistema se compone principalmente de tres partes: entrada, control de movimiento y salida.

McAdam et al. [17] discutieron la importancia que tiene el movimiento humano de forma natural para resolver problemas complejos en la dinámica del movimiento; por lo cual se indicó la importancia de representar un entorno virtual multisensorial donde el espacio de fase 3D del problema sea asignado a un sistema de coordenadas visuales en 3D, propusieron usar una computadora Dell T3400 de doble núcleo a 3.00 GHz, un monitor de 120Hz de 22 pulgadas y una pluma háptica Phantom Omni 6 grados de libertad para la entrada. La visualización 3D se implementó utilizando Microsoft DirectX en Windows 7 con renderización estereoscópica y efectos de sonido 3D para reforzar la posición y el movimiento de la bola en el espacio, la bola en el espacio 3D representa el estado actual del sistema. Como resultado final se obtiene la simulación de un sistema de bastón de carga.

Hayashi et al. [18] resaltaron las características de las interfaces naturales de usuario resaltando la modalidad de gestos físicos. Los autores indicaron la importancia de contar con un sistema que no requiera ninguna memorización, ya que a pesar de existir trabajos anteriores la capacidad de recordar estos gestos no está clara, por lo que su propuesta usa "Microsoft Kinect SDK 1.5", Kinect, Matlab, Weka, Macbook Pro con Intel Core i7 a 2,6 GHz. Se obtiene como resultado final la identificación del usuario basada en el cuerpo: una nueva forma de identificación

del usuario basada en las longitudes del segmento del cuerpo y un gesto de agitar la mano.

Heijboer et al. [19] resaltaron que el puntero de luz es un dispositivo adecuado para estudiar la viabilidad de las interacciones periféricas con una interfaz gestual. El objetivo de este experimento exploratorio fue rediseñar los estilos de interacción del puntero de luz de tal manera que potencialmente podría interactuar con la periferia de la atención. Por lo tanto, se exploró qué gestos y qué tipo de comentarios serían los más adecuados para la interacción periférica. El puntero de luz y una lámpara estuvieron controlados por un Phidget Interfacekit1 que está conectado a un Apple iMac ejecutando el software Max / MSP / Jitter.

De Carolis et al. [20] expresaron la importancia de las características *NUI (Natural User Interface, Interfaz Natural de Usuario)*, por lo tanto, el objetivo fue proporcionar a los usuarios una interfaz natural para acceder a información adicional sobre las prendas que se muestran en los maniqués desde el exterior, incluso cuando la tienda está cerrada. El prototipo se desarrolló utilizando un dispositivo comercial popular que permite la interacción sin contacto: Microsoft Kinect 21.

De acuerdo con Mehler et al. [21], resaltaron que cada vez más interfaces humano-computadora (HCI) son diseñadas para incorporar medios de comunicación no verbal. La escritura gestual se centra principalmente en el nivel perceptual de las descripciones de las imágenes. WikiNect se propuso como un sistema que permite autoría de hipertextos mediante escritura gestual. Para la propuesta se usó la tecnología Kinect y como es un prototipo no se mencionó algún lenguaje de programación, dividiendo su arquitectura en segmentación, vinculación, descripción y calificación de imágenes. El resultado final fue el módulo de gestión de sesión.

Barrón-Estrada et al. [22] discutieron la importancia de los entornos de aprendizaje tradicionales, los cuales no proporcionan un modelo de aprendizaje individual. Para superar este problema se propuso un sistema que utilice expresiones faciales y señales de EEG (*Electroencephalography Signals, Señales Electroencefálicas*) para el reconocimiento de la emoción. El sistema utilizó la diadema Emotiv Epc para registrar las señales de EEG del estudiante y Motor de Lógica Difusa, el cual es el responsable de evaluar emociones cognitivas y afectivas.

Wu et al. [23] destacaron la creciente popularidad del software de estimación de postura humana y reconocimiento espacial para el seguimiento automatizado del cuerpo humano y sobre todo el hecho de que los sistemas interactivos sufren cuando el objetivo rastreado es ocluido por otras personas u objetos desde el campo de visión del sistema. Por lo que es importante contar con una solución que resuelva el problema de oclusión para permitir a los sistemas interactivos rastrear consistentemente la información espacial (por ejemplo, posición) o fisiológica (por ejemplo, rasgos faciales) de los usuarios.

En [23] se mencionó la existencia de soluciones previas para la detección humana en imágenes, utilizando el seguimiento del esqueleto de Kinect, sin embargo, estos trabajos requieren que la mayoría de los usuarios permanezca sin obstrucciones desde el único campo de visión de la cámara, lo que limita el tipo de interacciones que se producirían normalmente. Se indicó la importancia de contar con un sistema para resolver el problema de oclusión, ya que las soluciones anteriores no lo resolvieron, por lo que los autores presentan *Out of Sight*, un sistema de seguimiento del esqueleto consciente de la oclusión, su propuesta usó los Kinects colocados en paralelo, con 45 y 90 grados de separación y una API RESTful para permitir el acceso distribuido de las posiciones de la articulación esquelética libre de oclusión, dividiendo el proyecto en tres etapas, una aplicación de detección, un procedimiento de calibración y un módulo de seguimiento. En primer lugar, la aplicación de detección procesa flujos de esqueleto entrantes de los Kinects. Entonces, la aplicación inicializa la calibración. Para cada esqueleto en cada campo de visión, calcula su posición central inicial, así como el ángulo entre su cuerpo y la cámara. Después de la calibración, las uniones del esqueleto de todos los campos de visión se transforman en un sistema de coordenadas mundial común, lo que permite una comparación entre los esqueletos, dando como resultado final el Kit de herramientas *Out of Sight*.

Kwan et al. [24] enfatizaron que las *GUI* (*Graphical User Interface*, Interfaces Gráficas de Usuario) de aplicaciones populares a menudo son inaccesibles para personas que se ven afectadas por trastornos o lesiones que causan problemas de movimiento severos, que no hablan y están imposibilitadas de usar sus manos para operar un *mouse* de computadora, por lo que es importante contar con una solución que contemple una interfaz adaptable que permita que las personas con impedimentos de movimiento severos, que usan dispositivos de entrada de ayuda, puedan interactuar con las *GUI* con éxito a través de barras deslizantes.

En [24] describieron soluciones existentes, sin embargo, estos trabajos no son aplicables a Windows, por lo que proponen técnicas de interacción que se usan con sistemas de sustitución del mouse y permiten la creación de GUI accesibles, su propuesta usa UI Automation que ofrece muchas mejoras sobre la accesibilidad activa, WindowsAPI para determinar si el elemento es un submenú o un elemento real que debe hacerse clic, C # para la codificación, dando como resultado dos sistemas: Camera Canvas que utiliza una barra de herramientas deslizante como su controlador de menú universal, los parámetros de la barra de herramientas se adaptan automáticamente a las capacidades de movimiento del usuario específico, y Menu Controller procesa automáticamente los menús de algunas aplicaciones existentes en barras de herramientas deslizantes adaptativas.

Tello et al. [25] remarcaron las características únicas que tienen las BCI (*Brain-Computer Interface*, Interfaz cerebro-computadora) y sobre todo el hecho de que es importante contar con una interfaz robusta, eficiente y funcional del tipo SSVEP (*Steady-State Visual Evoked Potentials*, Potenciales Evocados Visuales de

Régimen Estable)-BCI aplicada a una silla de ruedas que pueda navegar en un camino establecido o en un ambiente sin supervisión.

En [25] no destacaron soluciones anteriores aplicadas a SSVEP-BCI, por otra parte, los autores indicaron la importancia de contar con un dispositivo de alta precisión utilizando potenciales evocados visuales, y diferentes extractores de características, por lo que su propuesta usa *BrainNet-36* para la adquisición de las señales de *EEG (Electroencefalografía)*, y una silla de ruedas robotizada para comprobar las señales enviadas.

Cardoso et al. [26] resaltaron la importancia de la aplicación de tecnología para ayudar a las personas, por lo que es importante contar con una solución que contemple aliviar su incomodidad en una variedad de tareas, principalmente actividades de rutina, al proporcionar más autosuficiencia y reducir la dependencia de la ayuda externa a personas con discapacidades.

En [26] se indicó la importancia de contar con un dispositivo que permita satisfacer las necesidades de una comunicación inmediata y eficiente entre una persona con discapacidad motora y su cuidador, utilizando modernos recursos tecnológicos de seguimiento ocular y dispositivos móviles, por lo que presentan SMAD (Sistema para Domótica de Asistencia Multimodal) para que un usuario con discapacidades motoras sea capaz de controlar dispositivos del hogar desde una silla de ruedas por medio de señales biológicas capturadas de los músculos y de los ojos, su propuesta usa Emotiv EPOC para transmitir valores lógicos discretos al Controlador de dispositivo; Eye Tribe Tracker; una silla de ruedas robótica de UFES (*Federal University of Espirito Santo, Universidad Federal de Espirito Santo*), se incrustó una PC justo detrás del respaldo, conectada a una pantalla LCD ubicada frente al usuario; SmartBox que es una caja de acrílico externa a la que están conectados los accesorios del hogar; dos placas Arduino, la primera placa está conectada al puerto USB de la computadora de la silla de ruedas, y el módulo CI (*Control Interface, Interfaz de control*) interactúa con esta placa a través de la conexión USB, simulando una interfaz en serie, la segunda placa está instalada en la SmartBox; Bluetooth API se utilizó para permitir la comunicación con el módulo Bluetooth integrado al sistema y la telefonía ("android.telephony") para enviar mensajes de texto. El proyecto se divide en tres módulos, el Módulo BST (*Biological Signal Translator, Traductor de señales biológicas*) adquiere señales biológicas y las traduce en una señal de salida de control, el Módulo CI es responsable de traducir la salida BST en señales semánticas de control significativas para el controlador del dispositivo, el Módulo de controlador de dispositivo consta del hardware y el software que recibe el identificador de acción del CI y transmite la información al SmartBox. Finalmente, se desarrolló una aplicación móvil, llamada CAApp, para el sistema operativo Android para recibir comandos de interfaz en el teléfono inteligente, en conjunto se forma SMAD, un sistema de asistencia multimodal para controlar un entorno inteligente mediante señales biológicas o bioseñales.

2.2 Análisis comparativo

La Tabla 2.1 muestra un análisis comparativo entre los artículos relacionados que se obtuvieron para este proyecto, para mostrar mejor las diferencias y similitudes entre estos.

Tabla 2. 1. Análisis comparativo de los trabajos relacionados.

Artículo	Problema	Tecnologías	Resultados	Estado
[13]	No contar con un dispositivo que resulte medicamente valioso para ayudar a personas con problemas de parálisis	Diadema Emotiv EPOC, Scratch, lenguaje de programación Python, robot Finch, sensores de luz, temperatura, obstáculos, Motores, zumbadores, acelerómetros, pico de color, lápiz para dibujar y puerto USB.	Se encontró que la diadema Emotiv EPOC tuvo éxito al ser conectada y controlar al robot Finch. Con algunas fallas, el dispositivo reconoció los movimientos de personas y los tradujo en pulsaciones de teclado que luego se enviaron con éxito como comandos a los programas Scratch y Python.	Finalizado
[14]	Interfaces gestuales con precio elevado y calibración personalizada para cada usuario.	Acelerómetro DXL345, microntrolador ATmega328A, <i>driver</i> FTDI232RL, placa de circuito impreso, cinta elástica, Bluetooth (módulo RN41-XV), Labview.	La posibilidad del control desde diferentes partes del cuerpo, y la adaptación a cada usuario mediante el ajuste de los parámetros de procesado y calibración de la interfaz gestual desarrollada, así como su bajo costo hace que sea utilizada por un gran número de usuarios potenciales. Además la posibilidad de realizar un control cooperativo de las diferentes aplicaciones permite aumentar la integración social de los usuarios.	Finalizado

Tabla 2.1 Análisis comparativo de los trabajos relacionados.

Artículo	Problema	Tecnologías	Resultados	Estado
[15]	Método que registre la orientación de objetos en movimiento.	Pantalla grande (89x50 cm, 1920x1080px) y un seguidor ocular remoto Tobii X300.	Las tres aplicaciones presentadas en este trabajo muestran que el método Pursuits es natural y eficaz para la selección de objetivos móviles y permite aprovechar el movimiento en las interfaces de usuario de maneras innovadoras.	Mejoras a futuro
[16]	Contar con una solución de tecnologías de seguimiento de manos y reconocimiento de gestos para simplificar las operaciones y proporcionar una interfaz intuitiva.	Controlador Leap Motion, Leap Motion SDK, Maya, Unity3D Pro V4.	La usabilidad del sistema se examinó preliminarmente en una prueba, y los resultados mostraron que los niños pequeños se benefician jugando con Puppet Narrator, los resultados del análisis son prometedores. Sin embargo, dado que solo se evalúa un número limitado de sujetos, se necesitará examinar más casos y diseñar un experimento psicológico para reafirmar estos resultados.	Mejoras a futuro
[17]	Aprovechar la capacidad natural para aprender nuevas habilidades de movimiento de tal forma que el usuario explore, comprenda y controle sistemas arbitrarios caracterizados por dinámicas no lineales.	Computadora Dell T3400, un monitor de 120Hz de 22 pulgadas y una pluma háptica Phantom Omni 6 grados, Microsoft DirectX en Windows 7.	Se comprobó la efectividad de un entorno simple de CIS para involucrar las capacidades sensorio-motoras de un usuario en la exploración de un sistema dinámico no lineal.	Finalizado

Tabla 2.1 Análisis comparativo de los trabajos relacionados.

Artículo	Problema	Tecnologías	Resultados	Estado
[18]	Contar con un sistema de identificación de usuario que no requiera ninguna memorización.	Microsoft Kinect SDK 1.5, Kinect, Matlab, Weka. Macbook Pro con Intel Core i7 a 2,6 GHz.	El estudio de campo demostró que el sistema propuesto funcionaba bien en situaciones ecológicamente válidas, en términos de diferentes entornos y oclusiones parciales.	Finalizado
[19]	Evaluar las capacidades de atención necesarias para interactuar gestualmente con un sistema de luces interactivo.	Phidget Interfacekit1, Apple iMac software Max / MSP / Jitter.	Los resultados mostraron que el rendimiento con el puntero de luz, mientras realizaba la tarea de alta atención simultáneamente, era mejor en varias mediciones cuando la retroalimentación combinaba múltiples modalidades.	Mejoras a futuro
[20]	Acceder a información adicional sobre las prendas que se muestran en los maniqués desde el exterior, incluso cuando la tienda está cerrada.	Microsoft Kinect 21.	La interacción basada en la mirada resultó apropiada para la tarea, así como natural e interesante, mejorando así la experiencia del usuario.	Mejoras a futuro.
[21]	No contar con un sistema que permita autoría de hipertextos mediante escritura gestual.	Tecnología Kinect.	Proporcionó un punto de partida para profundizar esta visión HCI común en una perspectiva semiótica.	Mejoras a futuro.
[22]	Las necesidades afectivas y cognitivas para proporcionar enseñanza individualizada a los estudiantes que aprenden un lenguaje de programación.	Diadema Emotiv Epc, Motor de Inferencia Fuzzy.	Se logró enseñar individualmente a los estudiantes con conocimientos previos del lenguaje de programación Java, basándose en su estado cognitivo y afectivo.	Mejoras a futuro.

Tabla 2.1 Análisis comparativo de los trabajos relacionados

Artículo	Problema	Tecnologías	Resultados	Estado
[23]	El problema de oclusión en sistemas de seguimiento e identificación en tiempo real de las posiciones de las articulaciones de las personas es común en la implementación real de sistemas de cámara frontal única. Durante la oclusión, el sistema no ubica las posiciones de las articulaciones corporales de los usuarios.	Kinect, RESTful.	El sistema rastrea consistentemente a la persona cuando camina alrededor de un obstáculo, al probar la API y el kit de herramientas con 10 participantes para rastrear personas ocluidas, los esqueletos se rastrearon correctamente.	Mejoras a futuro.
[24]	Las interfaces gráficas de usuario del software popular a menudo son inaccesibles para personas con impedimentos graves de movimiento, que no pueden usar el teclado y el <i>mouse</i> tradicionales, y requieren un dispositivo de entrada alternativo.	UI Automation, WindowsAPI, C #.	El resultado más importante fue el concepto de actualización de GUI a través de barras deslizantes.	Mejoras a futuro.

Tabla 2.1 Análisis comparativo de los trabajos relacionados.

Artículo	Problema	Tecnologías	Resultados	Estado
[25]	Contar con una robusta, eficiente y funcional SSVEP-BCI aplicada a una silla de ruedas que pueda navegar en un camino establecido o en un ambiente sin supervisión.	BrainNet-36, silla de ruedas robotizada.	La técnica de MSI consiguió los más altos valores de acierto, con un promedio de 88%. Asimismo, se confirmó que la técnica de MSI demostró ser la más estable debido a su baja desviación estándar.	Mejoras a futuro.
[26]	Contar con un dispositivo para domótica de asistencia multimodal que tenga facilidad de configuración y uso, bajo costo y personalización.	Emotiv EPOC Eye Tribe Tracker, silla de ruedas robótica, pantalla LCD, placas Arduino, Bluetooth API, android.telephony.	La precisión promedio y la duración de la prueba indicaron reconocimiento de gestos bastante rápido y confiable.	Mejoras a futuro

Cada vez es más la necesidad de las personas de interactuar en la vida diaria con interfaces humano-computadora ya sea para ocio, apoyo a la vida diaria o en negocios, lo que ocasiona la necesidad de desarrollar aplicaciones que contengan el manejo de interfaces humano-computadora eficientes a las que tengan acceso más fácilmente las personas.

Los trabajos analizados se basan en sensores, Kinect, Emotiv Epoc para la recopilación de señales cerebrales. No se encontró reporte de la existencia de herramientas que apoyen el desarrollo de este tipo de aplicaciones, lo que justifica el trabajo a desarrollar.

2.3. Propuesta de la solución

Se eligió Emotiv Epoc + por su característica de reconocer expresiones faciales y su SDK brinda las herramientas que se necesitan para administrar y escalar aplicaciones, Java se escogió por su buena integración con el SDK de Emotiv Epoc+, todo esto bajo el entorno de desarrollo NetBeans, tomando en cuenta el tamaño del equipo que participa y el tiempo de desarrollo, de tal manera se eligió Scrum.

2.3.1. Justificación de la tecnología seleccionada

La decisión está determinada por la conjugación de las tecnologías de la información descritas a continuación:

- **Metodología:** Por los tiempos de desarrollo se seleccionó una metodología que se rige por un enfoque ágil de desarrollo de software, de tal manera se optó por SCRUM, dado que se centra en la entrega final del producto de software de forma iterativa e incremental y el hecho de no ignorar la documentación por completo, dando flexibilidad para inclusión de los artefactos necesarios para que la ingeniería de software sustente los requerimientos y características de la solución dada.
- **Lenguaje de programación:** Java es un lenguaje de programación orientado a objetos, de alto nivel que al mismo tiempo es compilado e interpretado. El cual por la naturaleza del proyecto cuenta con las bibliotecas necesarias para la parte gráfica que se necesita desarrollar e implementa naturalmente el patrón arquitectónico MVC, dado que el modelo de objetos es más simple y elimina las herramientas de bajo nivel y fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

- **Dispositivo:** Se eligió Emotiv EPOC +, dado que es un electroencefalograma multicanal y está diseñado para la investigación del cerebro humano escalable y contextual y aplicaciones avanzadas de interfaz cerebro-computadora y brinda acceso a datos cerebrales de grado profesional, su plataforma de desarrollo es compatible con Java y a diferencia de otros cuenta con reconocimiento de expresiones faciales (parpadeo, guiño, sorpresa, fruncir el ceño, sonreír, apretar, reír, sonreír) dada la naturaleza del proyecto será muy necesario.
- **Entorno de desarrollo:** Se eligió NetBeans debido a su gran interfaz de desarrollo y la gran utilidad con la que cuenta para hacer aplicaciones gráficas.

Capítulo 3. Aplicación de la metodología

El objetivo principal de este capítulo es dar a conocer el desarrollo de la solución, siguiendo las reglas de la metodología Scrum la cual fue elegida para el presente proyecto.

3.1. Descripción de la solución

En la presente sección se mostrará detalladamente la implementación de la solución propuesta mediante Scrum.

Primeramente, se realizaron reuniones con la directora del proyecto la M.C. María Antonieta Abud Figueroa la cual expuso los elementos principales que debe tener la biblioteca de interfaces gestuales y cómo debe de ser su funcionamiento.

Después de acordar los requerimientos funcionales que debe incluir la biblioteca de interfaces gestuales se hizo la adaptación a la metodología Scrum. Definido ya el alcance se prosiguió con la construcción de la pila del producto la cual se muestra en la Tabla 3.1.

Tabla 3. 1. Pila del producto.

No.	Descripción
1	La biblioteca debe proporcionar funcionalidades que permitan incluir control del <i>mouse</i> a través de gestos faciales.
2	Los controles del <i>mouse</i> a incluir los movimientos: arriba, abajo, izquierda, derecha y clic.
3	Para la identificación de los gestos faciales se utilizará la diadema Emotiv Epoc +.
4	Los gestos faciales a utilizar son: parpadeo derecho, parpadeo izquierdo, sonrisa, mirada hacia la derecha y mirada hacia la izquierda.
5	El usuario de la biblioteca podrá configurar qué gesto se asocia a qué función del <i>mouse</i> .
6	La biblioteca deberá funcionar en aplicaciones gráficas desarrolladas en lenguaje Java a partir de la versión 8.

Una vez establecida la pila del producto, se decidió establecer 6 Sprints. Algunos por la naturaleza del proyecto no inciden directamente en la pila del producto, pero son necesarios para cumplir los objetivos. A continuación, se muestran las iteraciones (Sprints) planeadas:

1. Análisis del dispositivo y su *suite* de desarrollo para la recepción de señales EEG.
2. Especificación de comandos que se incorporarán a la biblioteca que servirán para desarrollar interfaces gestuales.
3. Diseño de la arquitectura de la Biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.
4. Desarrollo de la biblioteca.
5. Incorporación de la biblioteca en el ambiente Java.
6. Aplicación a un caso de estudio.

3.1.1. *Sprint* 1 Analizar dispositivo y su *suite* de desarrollo para la recepción de señales EEG.

Las actividades realizadas durante el primer Sprint se muestran en la Tabla 3.2.

Tabla 3. 2..Sprint número uno.

Número de Sprint	Nombre	Tareas
1	Análisis del dispositivo y su <i>suite</i> de desarrollo para la recepción de señales EEG	Analizar el dispositivo Emotiv Epoc +
		Analizar <i>suite</i> de desarrollo de Emotiv Epoc +
Propósito	Conocer el funcionamiento correcto de la diadema Emotiv Epoc + así como su <i>suite</i> de desarrollo.	
Fecha	7/enero/2019 al 28/Febrero/2019	
Estado	Finalizado.	

Para el inicio de este *Sprint* inicialmente se investigó el uso correcto de la diadema, el cual consiste en cargar completamente la batería y colocar los electrodos, con lo cual quedará como se muestra en la Figura 3.1.



Figura 3. 1. Diadema Emotiv Epoc + con electrodos colocados.

Antes de utilizar la diadema, los electrodos deben bañarse con una solución salina, no se deben quitar necesariamente de la diadema para humectarse, la solución recomendada es una solución salina utilizada en los lentes de contacto.

Después de analizar la parte física del dispositivo se descargó e instaló el software EmotivBCI el cual es software principal para la interfaz cerebro-computadora, creada para Insight y Epoc +. Éste contiene opciones de entrenar expresiones faciales como son sorpresa, fruncir el ceño, sonreír, entre otros. El entrenamiento de las expresiones faciales se realiza en vivo mediante un avatar el cual indica la expresión facial que se debe realizar con lo cual se contabilizan para que a la hora de vincular la diadema reconozca claramente tales expresiones. La página de descarga se muestra en la Figura 3.2. Para utilizar la diadema con la computadora es necesario conectar primeramente el módulo de Bluetooth en el puerto USB, de tal manera será reconocida la diadema.

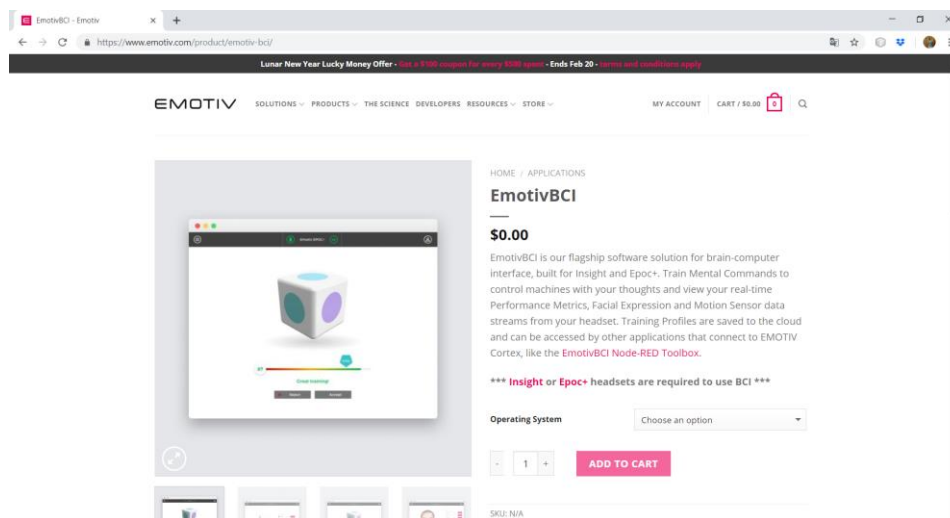


Figura 3 2.Página de descarga de EmotivBCI

Después de realizar el entrenamiento facial como se muestra la Figura 3.3, se investigaron las clases para conectar el SDK de Emotiv con el lenguaje de programación Java, las cuales están disponibles en la comunidad oficial de Emotiv en *github.com*, las cuales se muestran en la Tabla 3.3. Estas clases sirven para programar errores, utilizar la nube y manejar estados mentales.

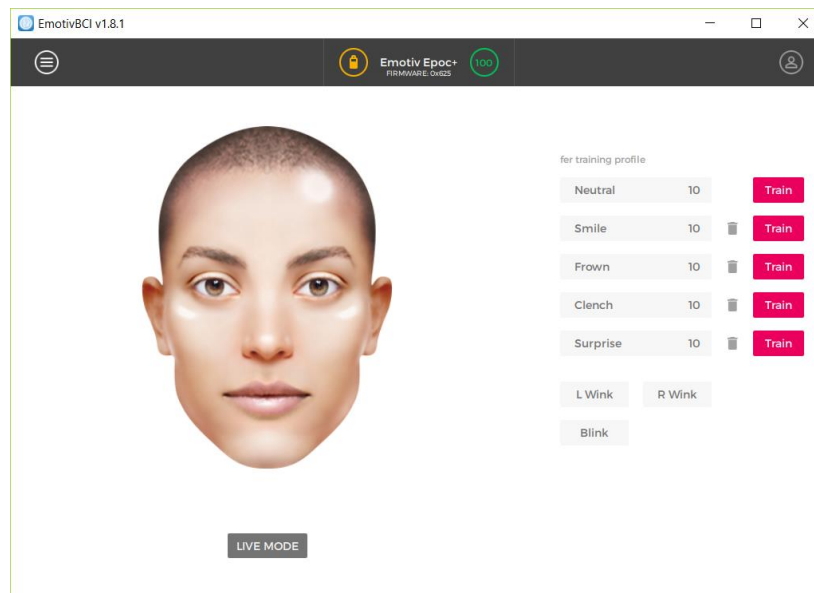


Figura 3.3. Apartado de entrenamiento de expresiones faciales en EmotivBCI.

Tabla 3.3. Clases de SDK de Emotiv y su funcionamiento.

Clase	Funcionamiento
Edk.java	Clase que contiene la información necesaria para el manejo de la diadema con Java.
EdkErrorCode.java	Contiene los errores relacionados con el manejo de la diadema.
EmoState.java	Contiene información para el uso de Java con expresiones faciales, comandos mentales y métricas.
EmotivCloudClient.java	Clase para trabajar con la diadema Emotiv y Java en la nube.
EmotivCloudErrorCode.java	Clase para errores relacionados con el manejo de la diadema y la programación en la nube.

Para verificar el uso de las clases se desarrolló un pequeño programa ejemplo en el cual se imprime en consola el nombre de la expresión facial que se realiza. Este ejemplo es el parte aguas para definir la estructura de la biblioteca. Como parte del

resultado del análisis de la diadema se encontró que cuando se cuenta con abundante cabello tiene la diadema dificultades en reconocer los electrodos, igualmente si no están bien hidratados. Con estos resultados satisfactorios se terminó el *Sprint* en tiempo y forma.

3.1.2. *Sprint* 2 Definir los comandos que se incorporarán a la biblioteca que servirán para desarrollar interfaces gestuales.

Las actividades realizadas en este *Sprint* se muestran en la Tabla 3.4.

Tabla 3. 4. *Sprint* número dos.

Número de Sprint	Nombre	Tareas
2	Especificación de comandos que se incorporarán a la biblioteca que servirán para desarrollar interfaces gestuales	Analizar gestos faciales disponibles con la diadema Emotiv epoc +.
		Realizar pruebas con los gestos faciales.
		Definir gestos faciales que se incorporarán a la biblioteca.
		Selección de movimientos que se incorporarán a la biblioteca los cuales se generarán mediante las expresiones faciales.
Propósito	Analizar y definir los comandos con los cuales contará la biblioteca de interfaces gestuales.	
Fecha	7/enero/2019 al 29/Marzo/2019	
Estado	Finalizado.	

En el segundo *Sprint* se inició realizando pruebas para analizar los gestos faciales disponibles para trabajar con la diadema Emotiv Epoc +, se realizaron las pruebas pertinentes con el programa donde se muestra el nombre del gesto facial cuando se realiza tal gesto.

Se realizaron las pruebas pertinentes con los gestos faciales como se muestra en la Figura 3.4, para ver cómo responden los gestos faciales mediante la conexión de la diadema Emotiv Epoc + con Java.



Figura 3 4. Prueba de gestos faciales Guiño Derecho.






Durante las pruebas se realizó una aplicación en lenguaje Java Swing que muestra cómo al accionar un gesto facial se activa un botón dentro de la aplicación. La Figura 3.5 muestra que al realizar guiño izquierdo se activa el botón dos y coloca el nombre en el cuadro de texto.



Figura 3 5. Aplicación muestra para verificación de funcionamiento de gestos faciales.



Realizadas las pruebas pertinentes se definieron los gestos faciales que se incorporarán a la biblioteca de interfaces gestuales los cuales se muestran en la Tabla 3.5.

Tabla 3. 5. Gestos faciales definidos para la biblioteca de interfaces gestuales.

Nombre	Gesto
Left wink (Guiño Izquierdo)	
Right Wink (Guiño Derecho)	
Smile (Sonreír)	
Looking Left (Mirada izquierda)	
Looking Right (Mirada Derecha)	




Como resultado de las pruebas con los gestos faciales algunos fueron descartados tales como guiño y apretar la cara (ver Tabla 3.6), ya que involuntariamente al realizar estos gestos se mueven músculos faciales, tales movimientos involuntarios son reconocidos como gestos faciales lo cual afecta al funcionamiento de las aplicaciones.


Tabla 3. 6. Gestos faciales descartados para el desarrollo de la biblioteca de interfaces gestuales.

Nombre	Gesto
Wink (Guiño)	
Clench (Apretar)	

Otro aporte en los resultados fue la definición de los movimientos que actuarán dentro de la biblioteca, los cuales se eligieron debido a que simulan los movimientos de *ratón* de la computadora. Dichos movimientos se muestran en la Tabla 3.7.

Tabla 3. 7. Definición de los movimientos que actuarán dentro de la biblioteca de interfaces gestuales.

Nombre	Acción
Arriba	
Abajo	
Izquierda	

Nombre	Acción
Derecha	
Clic Izquierdo	

3.1.3. *Sprint* 3. Diseñar la arquitectura de la biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.

Las actividades realizadas en este Sprint se muestran en la Tabla 3.8.

Tabla 3. 8. Sprint número tres.

Número de Sprint	Nombre	Tareas
3	Diseño de la arquitectura de la biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.	Analizar arquitecturas de software.
		Definir arquitectura.
Propósito	Diseñar la arquitectura de la biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.	
Fecha	4/marzo/2019 al 30/abril/2019	
Estado	Finalizado.	

Para iniciar el diseño de la arquitectura con la que trabajará la biblioteca, primeramente, se analizaron arquitecturas que trabajan con Java Swing. Como resultado se determinó utilizar el patrón arquitectónico MVC ya que éste tiene una gran influencia en el desarrollo de diseño de interfaces gráficas de usuario. Su objetivo principal es promover la independencia de desarrollo de sus componentes ya que si se desea hacer un cambio en algún componente no afectará a los demás y viceversa.

El componente modelo encapsula la información utilizada por la aplicación; el componente vista encapsula la información elegida y necesaria para la representación gráfica de esa información; el componente controlador encapsula la lógica necesaria para mantener el buen funcionamiento entre el modelo y la vista y maneja los eventos de entrada del usuario [27]. Tomando esto como base se realizó el diseño de la arquitectura la cual se muestra en la Figura 3.6.

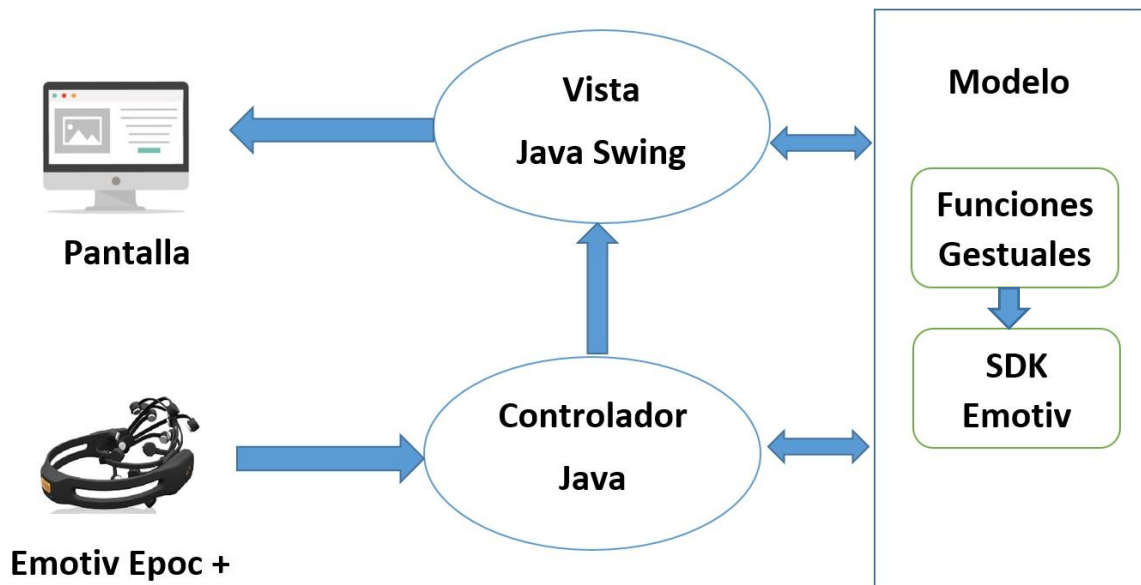


Figura 3.6. Arquitectura Patrón de arquitectura Modelo Vista Controlador.

1. **Vista:** Este componente presenta la interfaz gráfica la cual en este caso es Java Swing, interfaz que será controlada por medio de gestos faciales.
2. **Controlador:** Este componente contiene la lógica necesaria para la interacción entre el modelo y la vista, así como para manejar las interacciones del usuario.
3. **Modelo:** Componente que contiene la definición de las funciones que permitirán el control del apuntador de ratón a través de gestos faciales como son sonrisa, guiño derecho, guiño izquierdo, mirada derecha y mirada izquierda y se integran los flujos de datos al dispositivo y proporciona acceso a la API del mismo.

Para conectar el SDK de Emotiv a la biblioteca se utilizó el patrón de diseño Adapter como se muestra en la Figura 3.7.

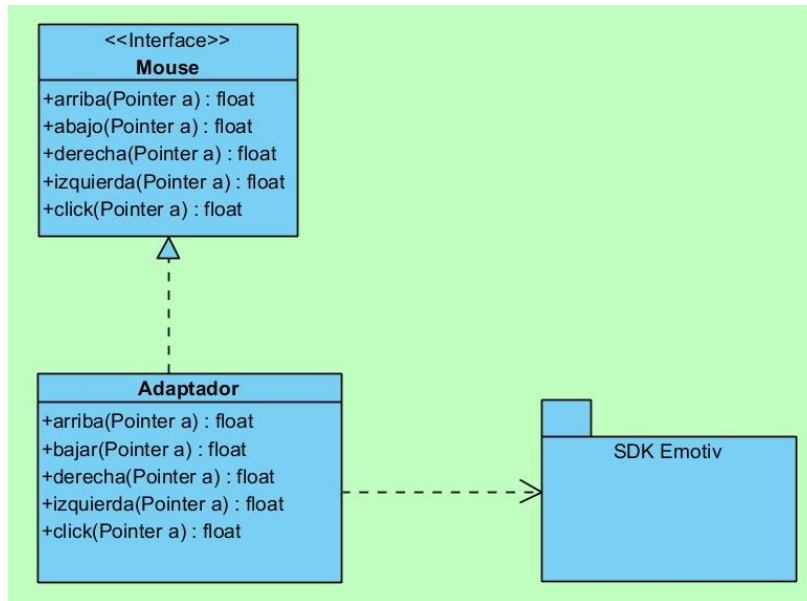


Figura 3 7. Patrón de diseño Adapter.

3.1.4. *Sprint* 4. Diseñar la arquitectura de la biblioteca para el desarrollo de interfaces humano-computadora utilizando expresiones faciales.

Las actividades realizadas en este Sprint se muestran en la Tabla 3.9

Tabla 3. 9. Sprint número cuatro.

Número de Sprint	Nombre	Tareas
4	Desarrollo de la capa de funciones gestuales.	Desarrollar el código
		Desarrollo de la capa de funciones gestuales en lenguaje de programación.
Propósito	Desarrollar la capa de Funciones Gestuales en lenguaje de programación Java.	
Fecha	30/abril/2019 al 28/Junio/2019	
Estado	Finalizado.	

Para el inicio se desarrolló la clase *Conexión* la cual servirá para realizar la conexión con Java y el SDK para el desarrollo de la biblioteca de interfaces gestuales, sin esta clase no se podrá detectar la diadema Emotiv Epoc +, el código de la clase se muestra en la Figura 3.8.

```

1  package model;
2
3  import sdk.Edk;
4  import sdk.EdkErrorCode;
5
6  /*
7   * To change this license header, choose License Headers in Project Properties.
8   * To change this template file, choose Tools | Templates
9   * and open the template in the editor.
10 */
11 /**
12  *
13  * @author INSPIRON
14  */
15 public class Conexion {
16
17
18     public void conectar() {
19
20         if (Edk.INSTANCE.IEE_EngineConnect("Emotiv Systems-5") != EdkErrorCode.EDK_OK
21             .ToInt()) {
22             System.out.println("Diadema no Conectada.");
23         }
24     }
25 }
26

```

Figura 3 8. Clase Conexion.

La clase Conexion cuenta con un método llamado conectar el cual contiene una condición en la que indica si la diadema está o no conectada. Haciendo referencia a la carga del SDK de Emotiv inicializando una conexión a una instancia remota de EmoEngine.

Tomando en cuenta el resultado del análisis de los movimientos de los ejes de las X y de las Y para los movimientos asignados durante el *Sprint 2*, se realizó el desarrollo de la clase *Apuntador*, la cual se muestra en la Figura 3.9. Cabe destacar que esta clase es muy importante para el funcionamiento de la biblioteca, puesto que con ella se logra la interacción con los gestos faciales.


```

Apuntador.java x
Source History
public class Apuntador extends JFrame {
21
22     private int y = 450;
23     private int x = 450;
24
25     @Override
26     public int getY() {
27         return y;
28     }
29
30     public void setY(int y) {
31         this.y = this.y + y;
32     }
33
34     @Override
35     public int getX() {
36         return x;
37     }
38
39     public void setX(int x) {
40         this.x = this.x + x;
41     }
42
43     public void run(int x, int y) {
44         try {
45             int a = x;
46             int b = y;
47
48             Robot r = new Robot();
49             r.setAutoDelay(700);
50             r.mouseMove(a, b);
51         } catch (Exception e) {
52             e.printStackTrace();
53         }
54

```

Figura 3 9. Código de la Clase Apuntador.

Para generar un apuntador es necesario emular el plano cartesiano el cual está formado por dos rectas numéricas, una horizontal y otra vertical, éstas se cortan en un punto. La recta horizontal es llamada eje de las abscisas o de las equis (x), y la vertical, eje de las ordenadas o de las yes, (y); el punto donde se cortan recibe el nombre de origen.

El plano cartesiano tiene como finalidad describir la posición de puntos, los cuales se representan por sus coordenadas o pares ordenados.

Las coordenadas se forman asociando un valor del eje de las X y uno de las Y , respectivamente, esto indica que un punto se puede ubicar en el plano cartesiano con base en sus coordenadas, lo cual se representa como: $P(x, y)$.

Para localizar puntos en el plano cartesiano se requiere llevar a cabo el siguiente procedimiento:

1. Para localizar la abscisa o valor de x , se cuentan las unidades correspondientes hacia la derecha si son positivas o hacia la izquierda si son negativas, a partir del punto de origen, en este caso el cero.
2. Desde donde se localiza el valor de X , se cuentan las unidades correspondientes hacia arriba si son positivas o hacia abajo, si son negativas y de esta forma se localiza cualquier punto dadas sus coordenadas [28].

Para emular el plano cartesiano con la clase Apuntador se contemplaron dos constantes de tipo entero las cuales son X y Y , éstas son constantes porque siempre iniciarán en los mismos ejes de la pantalla, se crearon sus métodos `get` y `set`. En los métodos `set` de las X y Y se recibirán el valor a avanzar en el mismo eje al que se mande el valor, cuando el método `setY()` recibe en el eje de las Y un número positivo el puntero se moverá hacia arriba y cuando el método `setY()` reciba un número negativo el puntero se moverá hacia abajo. De igual manera en el eje de las X cuando el método `setX()` reciba un número positivo el puntero se moverá hacia el lado derecho, y si el método `setY()` recibe un número negativo el puntero se moverá hacia la izquierda.

Todo esto se efectuará mediante los métodos `arriba()`, `abajo()`, `izquierda()`, `derecha()`, `run()`, de los cuales se explicará su funcionamiento a continuación:

- **arriba:** el método `arriba` al ser invocado envía al método `setY()` un valor negativo el cual en este caso es menos diez, ya que con este número el puntero del *ratón* se moverá hacia arriba y para que el movimiento sea efectuado envía al método `run` el valor de los ejes X y Y obteniendo los valores mediante los métodos `getX()` y `getY()`.
- **abajo:** el método `abajo` cuando es invocado envía al método `setY()` un valor positivo el cual en este caso es diez, ya que con este número el puntero del

ratón se moverá hacia abajo y para que el movimiento sea efectuado envía al método `run()` el valor de los ejes X y Y obteniendo los valores mediante los métodos `getX()` y `getY()`.

- **izquierda:** el método `izquierda()` al ser invocado envía al método `setX()` un valor negativo el cual en este caso es menos diez, ya que con este número el puntero del *ratón* se moverá hacia la izquierda y para que el movimiento sea efectuado envía al método `run()` el valor de los ejes X y Y obteniendo los valores mediante los métodos `getX()` y `getY()`.
- **derecha:** el método `derecha` cuando es invocado envía al método `setX()` un valor positivo el cual en este caso es diez, ya que con este número el puntero del *ratón* se moverá hacia la derecha y para que el movimiento sea efectuado envía al método `run()` el valor de los ejes X y Y obteniendo los valores mediante los métodos `getX()` y `getY()`.
- **run:** el método `run` es el encargado del movimiento de puntero del *mouse*, implementa la clase `Robot` que se encuentra en el paquete `java.awt`, esta clase está disponible desde la versión J2SE 1.3, fue elaborada con el propósito de la automatización de pruebas y cualquier actividad que involucre el *ratón* sin la presencia o interacción física del usuario, de dicha clase es utilizado el método `setAutoDelay()` el cual establece el número de milisegundos que el robot duerme después de generar un evento, en este caso cuando avanza el puntero del *ratón* se pausa por 700 milisegundos, también se utiliza el método `mouseMove()` el cual mueve el puntero del *mouse* a las coordenadas dadas.

Para simular el clic izquierdo del *ratón* se creó el método `clic()` dentro de la clase `Apuntador` como se muestra en la Figura 3.10.

```

74 public void clic() {
75
76     Robot r;
77     try {
78         r = new Robot();
79         r.mousePress(InputEvent.BUTTON1_MASK);
80         r.mouseRelease(InputEvent.BUTTON1_MASK);
81     } catch (AWTException ex) {
82         System.out.println(ex);
83     }
84
85 }
86

```

Figura 3 10. Código del Método `clic()`.

En este método se utiliza una instancia de la clase Robot del paquete java.awt, con dicha instancia es posible utilizar los métodos propios de dicha clase, aquí se utilizó el método `mousePress()` con el cual se puede seleccionar el botón del *ratón* con el que se trabajará.

Para la obtención de las señales de los gestos faciales se creó otro paquete dentro de la biblioteca llamado principal, el cual está conformado por ocho clases y una interfaz abstracta como lo muestra la Figura 3.11.

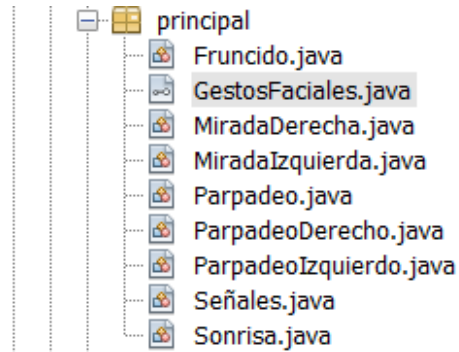


Figura 3 11. Contenido Paquete Principal.

La interfaz abstracta se implementó por todas las clases de gestos, ya que contiene la estructura principal para la obtención de los gestos faciales como se muestra en la Figura 3.12.

```

8
9
10 import com.sun.jna.Pointer;
11
12 /**
13  *
14  * @author INSPIRON
15  */
16 @
17 public interface GestosFaciales {
18
19     int numero1 = 1;
20     int numero2 = 2;
21     int numero3 = 3;
22     int numero4 = 4;
23     int numero5 = 5;
24     int numero6 = 6;
25     int numero7 = 7;
26
27     public int getNumero();
28
29     public float getGesto(Pointer a);
30 }
    
```

Figura 3 12. Código Interfaz GestosFaciales.

La interfaz tiene siete constantes, cada una será utilizada por cada clase en que es implementada, contiene dos métodos los cuales son llamados `getNumero()` que en cada clase obtendrá la constante a utilizar para la obtención de las señales de los gestos faciales, el método `getGesto()` es el encargado de obtener la señal de los gestos faciales haciéndolo mediante el patrón de diseño Adapter como se muestra en la Figura 3.13.

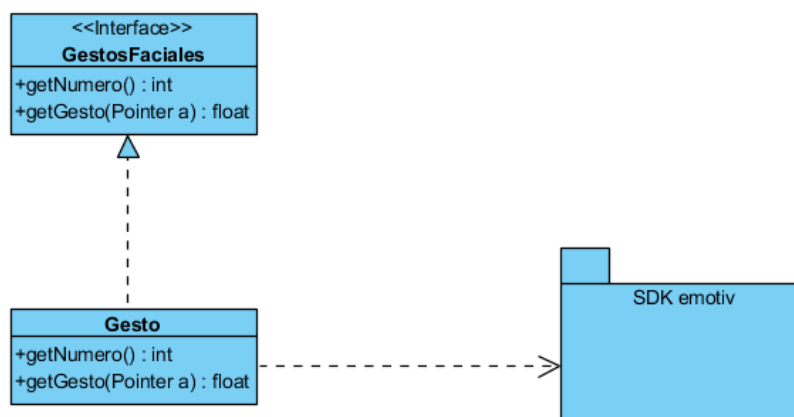


Figura 3 13. Representación del Patrón de diseño Adapter para la obtención de gestos faciales.

La primera clase que se describe es la clase `ParpadeoIzquierdo` la cual implementa la interfaz `GestosFaciales`, en el método `getNumero()` que retornará la constante `numero1` que fue declarada en la interfaz, el método `getGesto()` pregunta si el usuario está parpadeando con el ojo izquierdo en el momento en que `EmoState` está capturando y regresa el estado del guiño izquierdo (1: guiño, 0: no guiño), como se observa en la Figura 3.14.

```

10  import com.sun.jna.Pointer;
11  import sdk.EmoState;
12
13  /**
14   *
15   * @author INSPIRON
16   */
17  public class ParpadeoIzquierdo implements GestosFaciales{
18      @Override
19      public int getNumero() {
20          return numero1;
21      }
22
23      @Override
24      public float getGesto(Pointer a) {
25          return EmoState.INSTANCE.IS_FacialExpressionIsLeftWink(a);
26      }
27
28  }

```

Figura 3 14. Clase `ParpadeoIzquierdo`.

La siguiente clase a describir es `ParpadeoDerecho` la cual es la que obtiene la señal proveniente del parpadeo del ojo derecho, implementa la interfaz `GestosFaciales`, en el método `getNumero()` retornará la constante `numero2` la cual es de tipo entero que fue declarada en la interfaz, el método `getGesto()` pregunta si el usuario está parpadeando con el ojo derecho en el momento en que `EmoState` está capturando y regresa el estado del guiño derecho (1: guiño, 0: no guiño), la clase se muestra en la Figura 3.15.

```

17  public class ParpadeoDerecho implements GestosFaciales{
18
19
20      @Override
21      public int getNumero() {
22          return numero2;
23      }
24
25      @Override
26      public float getGesto(Pointer a) {
27          return EmoState.INSTANCE.IS_FacialExpressionIsRightWink(a);
28      }
29
30  }

```

Figura 3 15. Clase `ParpadeoDerecho`.

A continuación, son descritas las clases las cuales tienen el funcionamiento de adquirir las señales cuando los ojos miran hacia el lado izquierdo y hacia el lado derecho. Conseguir este tipo de señales es de gran importancia dentro de la biblioteca de interfaces gestuales ya que hace que tenga un repertorio más amplio de gestos faciales, con lo cual el programador tendrá diferentes opciones de acuerdo a sus necesidades, estas clases nombradas `MiradaIzquierda` y `MiradaDerecha` las cuales son descritas a continuación.

La clase `MiradaIzquierda` es la encargada de adquirir la señal que se origina cuando la mirada se dirige hacia el lado izquierdo, como clases anteriores que obtienen señales a partir de los gestos faciales implementa la interfaz `GestosFaciales`, el método `getGesto()` obtiene la señal del usuario cuando está mirando a la izquierda con auriculares EPOC + y devuelve la posición de los ojos (1: mirada Izquierda, 0: no mira a la izquierda), en el método `getNumero()` retornará la constante `numero4` la cual es de tipo entero que fue declarada en la interfaz, el código de la clase se observa en la Figura 3.16.

```

15  public class MiradaIzquierda implements GestosFaciales {
16
17      @Override
18      public int getNumero() {
19          return numero4;
20      }
21
22      @Override
23      public float getGesto(Pointer a) {
24          return EmoState.INSTANCE.IS_FacialExpressionIsLookingLeft(a);
25      }
26
27
28  }
```

Figura 3 16. Clase `MiradaIzquierda`.

Para la recepción de la señal de la mirada hacia la derecha es conseguida por la clase `MiradaDerecha`, en el método `getNumero()` retornará la constante `numero5` la cual como las demás contantes declaradas en la interfaz `GestosFaciales` es de tipo entero, el método `getGesto()` obtiene la señal del usuario cuando está mirando a la izquierda con auriculares EPOC + y devuelve la posición de los ojos (1: mirar a la derecha, 0: no mira a la derecha), la clase se muestra en la Figura 3.17.

```
17 public class MiradaDerecha implements GestosFaciales{
18     @Override
19     public int getNumero() {
20         return numero5;
21     }
22
23     @Override
24     public float getGesto(Pointer a) {
25         return EmoState.INSTANCE.IS_FacialExpressionIsLookingRight(a);
26     }
27
28 }
```

Figura 3 17. Clase MiradaDerecha.

Los gestos antes descritos son los que se realizan con los ojos. La ventaja de tener un repositorio de gestos faciales realizados a través de la vista es que muchas veces personas no pueden realizar algún movimiento y requieren interfaces especiales, con esta biblioteca el programador cuenta con más herramientas para desarrollar software orientado a personas con dificultad de movimiento. En las Figura 3.18 a 3.21 se ejemplifican estos gestos: mirada a la izquierda, mirada hacia la derecha, parpadeo izquierdo y mirada a la derecha.

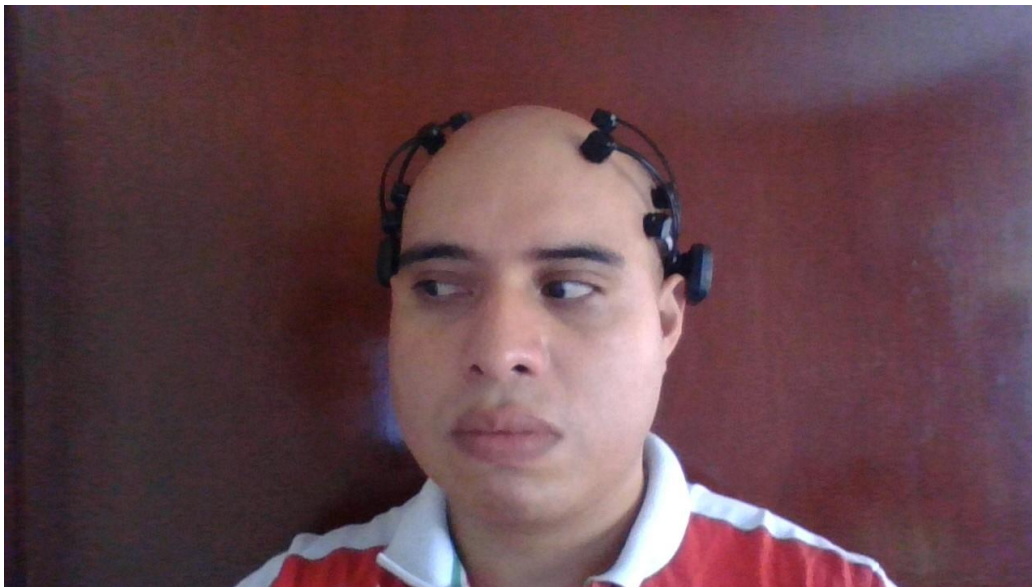


Figura 3 18. Mirada hacia la izquierda.



Figura 3 19. Mirada hacia la derecha.



Figura 3 20. Parpadeo izquierdo.



Figura 3 21. Parpadeo derecho.

Habiendo concluido con los gestos faciales que tienen relación con el movimiento de los ojos se tomó a consideración que en diversas ocasiones las personas no pueden realizar alguno de estos movimientos, por lo que para facilitar el uso de la biblioteca y con el objetivo de tener una amplia biblioteca de interfaces gestuales, se consideró incluir en ella el gesto de parpadeo que había sido excluido al principio, esto dará beneficios al proyecto tales como más herramientas al programador.

La clase Parpadeo, la cual adquiere la señal de los ojos cuando parpadean a través de la diadema Emotiv Epoc +, implementa la interfaz GestosFaciales, el método GetGesto() el cual hereda de la interfaz antes mencionada recibe una señal si el usuario parpadea en el momento en que EmoState está capturando y regresa el estado de parpadeo (1: parpadeo, 0: no parpadeo), el método getNumero() nos retornará la constante numero7 que fue declarada en la interfaz. El código de la clase se muestra en la Figura 3.22 y en la Figura 3.23 se muestra el ejemplo del uso de esta clase con la diadema Emotiv Epoc +.

```

15 public class Parpadeo implements GestosFaciales{
16     @Override
17     public int getNumero() {
18         return numero7;
19     }
20
21     @Override
22     public float getGesto(Pointer a) {
23         return EmoState.INSTANCE.IS_FacialExpressionIsBlink(a);
24     }
25
26 }

```

Figura 3 22. Clase Parpadeo.



Figura 3 23. Parpadeo.

Continuando con los gestos faciales incluidos dentro de la biblioteca de interfaces gestuales se tienen los gestos que se realizan con la boca, tales como son la sonrisa y los que implican nariz y boca como lo es el fruncido de cara, las cuales se describen a continuación.

Primeramente, se describe al gesto facial sonrisa la cual es un movimiento de la boca y sus músculos externos, para la recepción de la señal mediante la diadema Emotiv Epc + y Java se obtendrá mediante la implementación de la interfaz GestosFaciales con el método `getGesto()` al cual se le envía una variable de tipo Pointer y por medio del SDK de Emotiv devuelve la acción facial detectada, el

método `getNumero()` devuelve la constante `numero3` que es declarada en la interfaz. El código de la clase `Sonrisa` se muestra en la Figura 3.24.

```

17  public class Sonrisa implements GestosFaciales {
18
19
20
21      @Override
22      public int getNumero() {
23          return numero3;
24      }
25
26      @Override
27      public float getGesto(Pointer a) {
28
29          return EmoState.INSTANCE.IS_FacialExpressionGetLowerFaceActionPower(a);
30      }
31
32  }

```

Figura 3 24. Clase Sonrisa.

Siguiendo con la descripción de las clases de la biblioteca se continuó con el gesto facial de apretar los dientes, la clase se llama fruncido y como todas las clases que obtienen señales de los gestos faciales que son traídos del SDK de Emotiv implementa la interfaz `GestosFaciales`, la respuesta de la señal adquirida mediante el uso de la Diadema Emotiv epoc + se obtiene por medio del método `getGesto()` al cual se le envía una variable de tipo `Pointer`, el método `getNumero()` retorna la variable `numero6`, la clase se muestra en la Figura 3.25.

```

15  public class Fruncido implements GestosFaciales{
16      @Override
17      public int getNumero() {
18          return numero6;
19      }
20
21      @Override
22      public float getGesto(Pointer a) {
23
24          return EmoState.INSTANCE.IS_FacialExpressionGetLowerFaceAction(a);
25      }
26
27  }

```

Figura 3 25. Clase Fruncido.

Para terminar con los gestos faciales en la Figura 3.26 se muestra el gesto facial de la clase Fruncido.



Figura 3 26. Gesto Facial Fruncido.

Habiendo terminado las clases para obtener los gestos faciales y clases de Conexión y la clase `Apuntador`, se realizó la clase `Señales` la cual es la encargada de implementar todas las clases creadas las cuales son utilizadas para la recepción de las señales de los gestos faciales.

En la clase son creados objetos de la clase `ParpadeoDerecho`, `ParpadeoIzquierdo`, `MiradaDerecha`, `MiradaIzquierda`, `Fruncido`, `Parpadeo`, `Sonrisa`, sin olvidar la clase `Apuntar` y la clase `Conexion`, como se muestra en la Figura 3.27.

```

8  import com.sun.jna.Pointer;
9  import com.sun.jna.ptr.IntByReference;
10 import java.awt.AWTException;
11 import java.util.ArrayList;
12 import model.Apuntador;
13 import model.Conexion;
14 import sdk.Edk;
15 import sdk.EdkErrorCode;
16 import sdk.EmoState;
17
18 /**
19  *
20  * @author INSPIRON
21  */
22 public class Señales {
23
24
25     ParpadeoIzquierdo pi;
26     ParpadeoDerecho pd;
27     MiradaIzquierda mi;
28     MiradaDerecha md;
29     Sonrisa son;
30     Fruncido frun;
31     Parpadeo par;
32     Apuntador apunta;
33     Conexion con;

```

Figura 3 27. Clase Señales y creación de Objetos.

Después de crear los objetos de las clases realizadas para la obtención de las señales de los gestos faciales se crean las variables `st`, `usuario`, `opcion`, los objetos evento y estado los cuales servirán para manejar la elección de los gestos faciales dentro la biblioteca, estos se muestran en la Figura 3.28.

```

35     public int st = 0;
36     int opcion = 1;
37     IntByReference usuario = null;
38     Pointer evento = Edk.INSTANCE.IEE_EmoEngineEventCreate();
39     Pointer estado = Edk.INSTANCE.IEE_EmoStateCreate();
40

```

Figura 3 28. Variables y Objetos para el manejo de la biblioteca.

La clase `Señales` contiene dos métodos el primero es `resultado()`, que recibe una lista de enteros los cuales son los números de la posición de cómo serán configurados los gestos faciales, el segundo método es llamado `mouseDefault()` el cual tiene una configuración predeterminada del apuntador.

Primeramente, se creó el método `resultado()`, que en su firma no devuelve nada y recibe una lista de enteros los cuales servirán para la configuración del apuntador y su funcionamiento mediante interfaces gestuales.

Al entrar al método se realiza la creación de cinco variables de tipo entero a las cuales le son asignados los valores que trae la lista en las cinco posiciones que se recibe en la firma del método y que son los números de funcionamientos que tendrá el apuntador, todo esto se muestra en la Figura 3.29.

```

47 public void resultado(ArrayList<Integer> lista) throws AWTException {
48     System.out.println("Conectado");
49     int a = lista.get(0);
50     int b = lista.get(1);
51     int c = lista.get(2);
52     int d = lista.get(3);
53     int e = lista.get(4);

```

Figura 3 29. Firma del método resultado, creación y asignación de variables en el método.

Habiendo creado los variables que se asignan a la posición de las funciones del apuntador se implementan los objetos de las clases que fueron creadas para la recepción de las señales de los gestos faciales y esto se muestra en la Figura 3.30.

```

pi = new ParpadeoIzquierdo();
pd = new ParpadeoDerecho();
mi = new MiradaIzquierda();
md = new MiradaDerecha();
son = new Sonrisa();
frun = new Fruncido();
par = new Parpadeo();
con = new Conexion();
apunta = new Apuntador();
con.conectar();

```

Figura 3 30. Implementación de los objetos.

Se crea un ciclo `While` dentro del cual se manejan los eventos de los gestos faciales que va recibiendo del usuario con la diadema Emotiv Epoc + y va actualizando cada evento que es registrado como se aprecia en la Figura 3.31, una vez recibidos esos eventos la señal se procesa dentro de un `Switch` el cual tiene definidos el tipo de gesto y su funcionalidad como se observa en la Figura 3.32.

```

while (true) {
    st = Edk.INSTANCE.IEE_EngineGetNextEvent(evento);

    if (st == EdkErrorCode.EDK_OK.ToInt()) {

        int eventType = Edk.INSTANCE.IEE_EmoEngineEventGetType(evento);
        Edk.INSTANCE.IEE_EmoEngineEventGetUserId(evento, usuario);

        if (eventType == Edk.IEE_Event_t.IEE_EmoStateUpdated.ToInt()) {

            Edk.INSTANCE.IEE_EmoEngineEventGetEmoState(evento, estado);
        }
    }
}

```

Figura 3 31. Manejo de eventos de Gestos Faciales.


```

switch (a) {
  case 1:
    if (pi.getGesto(estado) == 1) {
      apunta.arriba();
    }
    break;
  case 2:
    if (pd.getGesto(estado) == 1) {
      apunta.arriba();
    }
    break;
  case 3:
    if (son.getGesto(estado) == 1) {
      apunta.arriba();
    }
    break;
  case 4:
    if (mi.getGesto(estado) == 1) {
      apunta.arriba();
    }
    break;
  case 5:
    if (md.getGesto(estado) == 1) {
      apunta.arriba();
    }
    break;
  case 6:
    if (frun.getGesto(estado) == 1) {
      apunta.arriba();
    }
    break;
}

```

Figura 3 32. Selección de funcionalidades de los Gestos Faciales.

El segundo método `mouseDefault()` es un método que tiene una funcionalidad por default, tales como parpadeo a la izquierda su funcionalidad siempre será subir el apuntador, parpadeo derecho será apuntador hacia abajo, mirada a la izquierda será el apuntador a la izquierda, mirada a la derecha será el apuntador a la derecha y la sonrisa será el clic, en el demás funcionamiento del método es semejante al método `resultado()`, el método `mouseDefault()` se muestra en la Figura 3.33.

```
public void mouseDefault() throws AWTException {
    int a = 1;
    int b = 2;
    int c = 4;
    int d = 5;
    int e = 3;
}
```

Figura 3 33. Método `mouseDefault()` .

3.1.5. *Sprint* 5. Incorporación de la biblioteca en el ambiente Java

Las actividades realizadas en este Sprint se muestran en la Tabla 3.10

Tabla 3. 10. Sprint número Cinco.

Número de Sprint	Nombre	Tareas
5	Incorporación de la biblioteca en el ambiente Java	Incorporar biblioteca en ambiente Java.
		Demostrar que se incorpora biblioteca en ambiente Java sin problemas.
Propósito	Incorporar la biblioteca de funciones gestuales en un ambiente Java para demostrar su funcionamiento.	
Fecha	29/Junio/2019 al 28/Julio/2019	
Estado	Finalizado.	

Una vez teniendo todas las clases se procedió a generar el archivo JAR el cual contendrá la biblioteca de interfaces, este archivo JAR será el utilizado por los programadores que requieran usar la biblioteca de interfaces gestuales, el archivo es nombrado BIG (*Biblioteca de Interfaces Gestuales*), el archivo `BIG.jar` se muestra en la Figura 3.34.

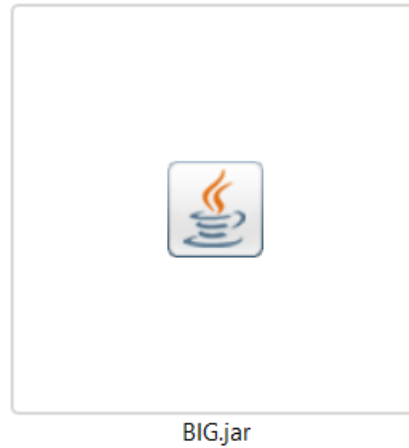


Figura 3 34. Archivo `BIG.jar`.

Ahora ya generado el archivo `BIG.jar` se mostrará un ejemplo de cómo debe de ser implementada dentro de algún proyecto en Java Swing que necesite de la biblioteca de Interfaces Gestuales.

Cuando se requiere usar la biblioteca de interfaces gestuales primeramente es necesario importar la biblioteca en el proyecto que está siendo utilizado, como se muestra en la Figura 3.35.

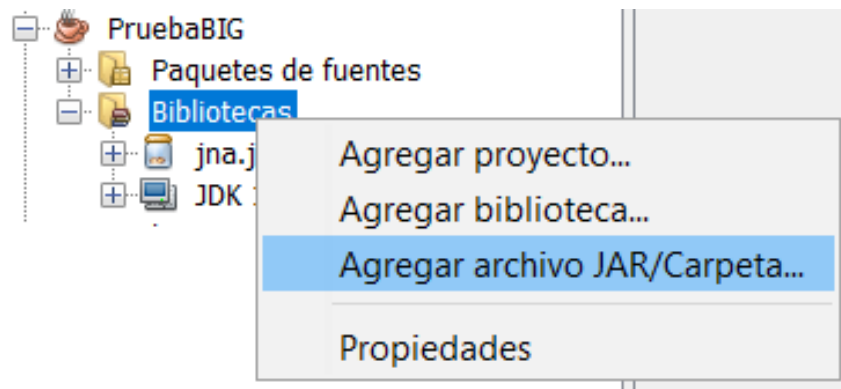


Figura 3 35. Importar archivo `BIG.jar`.

Se selecciona la ruta de donde este el archivo `BIG.jar`, Figura 3.36.

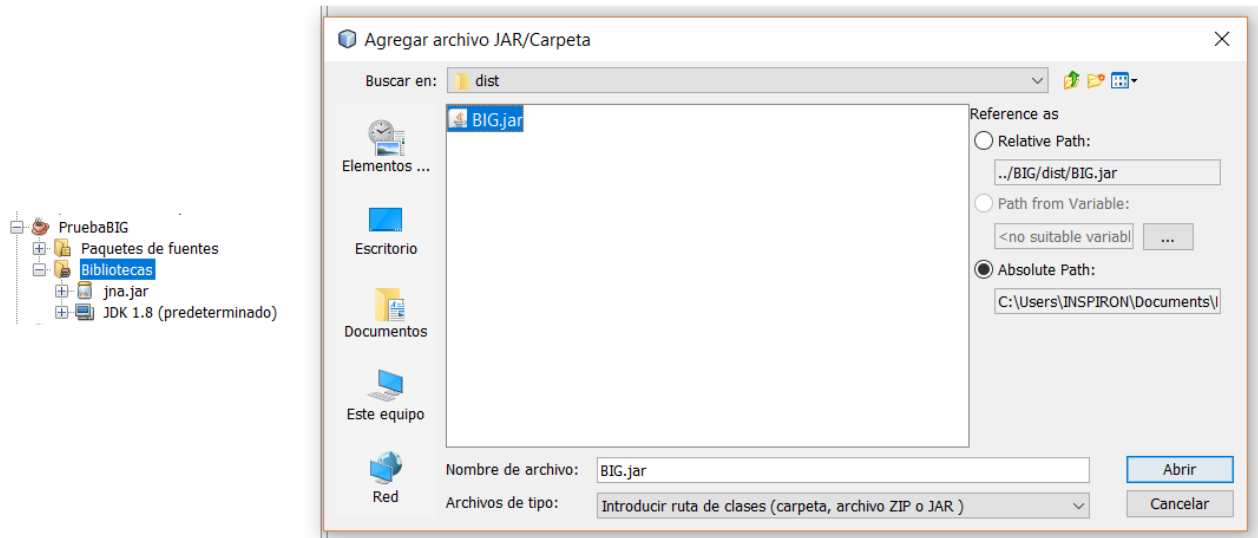


Figura 3 36. Selección de archivo `BIG.jar`.

El archivo `BIG.jar` ya cargado en el proyecto que necesita la biblioteca de Interfaces Gestuales, se muestra en la Figura 3.37.

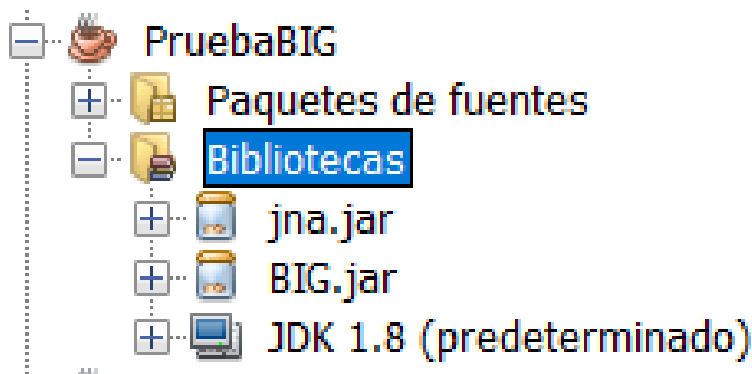


Figura 3 37. Archivo `BIG.jar` cargado en el proyecto.

Ya estando cargado el archivo `BIG.jar` en el proyecto que lo requiere se instancian las clases a utilizar dentro de clase principal dentro del método `main`, pero antes de eso se requiere importar la clase `Señales` dentro de la clase que funcione como

controlador dentro del proyecto en el cual se requiera el uso de la biblioteca de Interfaces Gestuales.

Lo primero que debe realizarse antes de importar las clases de Gestos Faciales que se utilizarán dentro de la clase principal en el método main, es dentro de la clase controlador del proyecto en el cual va a requerir la biblioteca de Interfaces gestuales es importar la clase Señales y crear un objeto de esta clase e instanciado dentro del controlador para seleccionar el método que se ocupará dentro de la biblioteca, el método el cual recibe el orden en el que serán utilizados los gestos faciales o el método que por *default* contiene una configuración predefinida. En la Figura 3.38 se observa la importación de la clase Señales, la creación de un objeto y su implementación en los constructores dentro de la clase Controlador del proyecto en el que se requiere la biblioteca de interfaces gestuales, con lo cual desde el método principal main se puede elegir si se envía una configuración a la necesidad del programador o se elige la configuración predefinida.

```

import java.awt.AWTException;
import java.util.ArrayList;
import view.View;
import principal.Señales;

/**
 *
 * @author INSPIRON
 */
public class Controller2 {

    View view;
    Señales señales = new Señales();

    public Controller2(ArrayList<Integer> lista) throws AWTException {
        this.view = new View(this);
        this.señales.resultado(lista);
    }

    public Controller2() throws AWTException {
        this.view = new View(this);
        this.señales.mouseDefault();
    }

}

```

Figura 3 38. Importación de Clase Señales de la Biblioteca de Interfaces Gestuales.

Como segundo paso se debe importar cada clase que se requiera utilizar de la biblioteca de Interfaces Gestuales, las clases para gestos faciales se encuentran en el paquete nombrado principal, se logra importar clase por clase o todo el paquete junto con el objetivo de lograr acceder a los métodos necesarios que estas clases contienen, la importación se muestra en la Figura 3.39.

```

import controller.Controller2;
import java.awt.AWTException;
import java.util.ArrayList;
import principal.*;

```

Figura 3 39. Importar Clases de la Biblioteca de Interfaces Gestuales.

Habiendo importado la clase `Señales` dentro de la clase que funcione como Controlador y las clases o el paquete de clases que son requeridas por el programador para el manejo de la biblioteca de interfaces gestuales dentro de la clase principal del proyecto donde se encuentre el método principal `main`, se crea una lista de enteros a la cual más adelante se le serán asignados los valores en las posiciones, pero antes que eso son creados e instanciados los objetos de las clases de los gestos faciales, antes de asignar la posición de cada gesto dentro de la lista se debe conocer que la configuración de las posiciones es la siguiente posición 1 arriba, posición 2 abajo; posición 3 derecha; posición 4 izquierda; posición 5 clic; ya conocida la configuración se le asignan las posiciones de la lista si es que se requiere tener una configuración propia del programador, la configuración se muestra en la Figura 3.40.

```
public static void main(String[] args) throws AWTException {
    ArrayList<Integer> arl = new ArrayList<>();
    ParpadeoIzquierdo pi = new ParpadeoIzquierdo(); //1
    ParpadeoDerecho pd = new ParpadeoDerecho(); //2
    MiradaIzquierda mi = new MiradaIzquierda(); //4
    MiradaDerecha md = new MiradaDerecha(); //5
    Sonrisa son = new Sonrisa(); //3
    Fruncido frun = new Fruncido(); //6
    Parpadeo par = new Parpadeo(); //7

    arl.add(par.getNumero());
    arl.add(frun.getNumero());
    arl.add(md.getNumero());
    arl.add(mi.getNumero());
    arl.add(son.getNumero());

    Controller2 e = new Controller2(arl);
}
```

Figura 3 40. Implementación de la Biblioteca de Interfaces Gestuales.

La creación de una biblioteca de interfaces gestuales es una gran herramienta para desarrolladores siendo lo más destacado la interacción humano-computadora sin necesidad de cables. En el siguiente capítulo se demostrará su funcionamiento en un caso de estudio.

Capítulo 4. Resultados

En este capítulo se muestran los resultados de la implementación de la biblioteca de interfaces gestuales, empezando por el caso de estudio, sus características y su eficacia.

4.1. *Sprint* 6. Aplicación a un caso de estudio

Las actividades realizadas en este Sprint se muestran en la Tabla 4.1

Tabla 4. 1. Sprint número Seis.

Número de Sprint	Nombre	Tareas
6	Aplicación a un caso de estudio.	Buscar donde aplicar la biblioteca de gestos faciales en un caso de estudio.
		Demostrar la funcionalidad de la biblioteca en un caso de estudio.
Propósito	Aplicación de la biblioteca de funciones gestuales en un caso de estudio para demostrar su funcionamiento.	
Fecha	29/Julio/2019 al 28/Octubre/2019	
Estado	Finalizado.	

4.1.1. Planteamiento del caso de estudio

Primeramente, se buscó a personas que tuvieran impedimento físico y que solo tuvieran movilidad de los gestos faciales, por lo cual se acudió al Centro de Rehabilitación Integral de Orizaba (CRIO) donde hicieron el comentario que encontrar personas de esa índole es algo complicado porque las personas que cuentan con esas características no entienden instrucciones, nos presentaron a una persona que cumplía las características necesarias, se le hicieron las pruebas con la diadema Emotiv Epoc + y resultó que los electrodos solo reconocían alrededor del 70 % de efectividad y sus gestos faciales no eran tan pronunciados como son requeridos, por lo cual se tomó la decisión de buscar otro tipo de personas que pudieran seguir instrucciones y mover libremente los músculos faciales. Por lo cual se acudió al Instituto de Psicología y Educación, centro de entrenamiento y educación especial de la Universidad Veracruzana con sede en Orizaba, Veracruz, donde cuentan con niños que sufren de déficit de atención con o sin hiperactividad, el objetivo de trabajar con esos niños es que sigan y respeten instrucciones para un mejor desarrollo. Se hicieron las pruebas de funcionamiento de la diadema Emotiv

Epoc + en los niños, las cuales salieron positivas en el reconocimiento de los electrodos y los gestos faciales, por lo tanto se planteó con la Mtra. Laura Patricia Medrano Herrera un sistema en el cual se implemente la biblioteca de interfaces gestuales y el cual será llamado "Implementación de biblioteca para el manejo de interfaces humano-computadora a través de gestos faciales en un Sistema de modificación de conducta con niños de déficit de atención con o sin hiperactividad", y el cual será descrito a continuación.

4.1.2. Introducción

Uno de los más graves problemas a que se enfrenta la Educación Especial es el manejo de los problemas de conducta en niños; es por esto que se plantea el siguiente programa que tiene como objetivo evaluar, diagnosticar y brindar tratamiento a las conductas por medio de los principios y procedimientos del Análisis Conductual Aplicado, teniendo como finalidad el desarrollo de repertorios sociales en niños que les permitan integrarse a su ambiente. Tomando en cuenta la dirección en la que se debe modificar la conducta del niño, dependiendo de las características de la misma. Para la implementación del programa es requisito indispensable la participación de los padres como co-terapeutas del tratamiento, ya que trabajarán directamente con el niño, siendo asesorados por el responsable del programa. De tal manera, se generará en los padres un manejo, control y dirección adecuada sobre el repertorio conductual de sus hijos.

A través de la historia los seres humanos se han formado dentro de un sistema social determinado por un amplio conjunto de influencias intrínsecas y extrínsecas que están determinando su comportamiento y que a su vez influyen y llevan a influir, formar y/o modificar el comportamiento de las personas que los rodean; es por esto que una gran parte de su comportamiento representa el resultado de lo han aprendido de los demás. Las personas se enseñan unas a otras a hablar, cuándo sonreír, qué ropa usar, cuándo hacer berrinche, cuándo pelear, cómo trabajar y toda una serie de comportamientos que se incluyen en diferentes rasgos de complejidad. Estas habilidades se aprenden accionando sobre los demás, por lo tanto, cuando dos o más personas interactúan se estimulan y modifican su comportamiento.

Con esto es fácil comprender que aprenden y enseñan a otras personas comportamientos socialmente adecuados, aunque es cierto también que las personas enseñan conductas inadecuadas sin siquiera percibirlo. A menos que observen lo que está ocurriendo, comprenden lo que le ocurre a nosotros y a las personas que los rodean. Sin embargo, la mayoría de la gente ignora los cambios que ocurren y que ellos están siendo agentes de cambio.

Ese proceso de interacción social se ve reflejado aún más dentro de la familia debido a que los padres enseñan a sus hijos muchas habilidades sociales, pero en ese proceso también se ven afectados por sus hijos.

Ya que se parte de la premisa que el niño es un organismo cuya sensibilidad está dispuesta a recibir la estimulación del ambiente en el cual inicia su desarrollo, es básicamente un ser receptivo durante los primeros años de su vida, ya que no cuenta con resistencias organizadas y eficaces para controlar su impresionabilidad y asimilación de estímulos externos. De lo anterior se desprende la importancia que las experiencias infantiles tienen en el desarrollo normal del individuo y en la formación de su personalidad posterior. La familia es el contexto ambiental en el cual se inicia el desarrollo infantil, y por lo tanto son los padres quienes juegan el papel más importante en la determinación de las adquisiciones conductuales del infante. Sin embargo, fácilmente se logra observar que los padres en la sociedad, carecen en su mayoría, de los recursos y la información necesaria para realizar adecuadamente sus tareas de educadores.

La psicología contemporánea, de acuerdo a su definición de ciencia que estudia la conducta humana en su más amplio sentido, ha establecido una serie de postulados y leyes sobre el aprendizaje que rige la adquisición de la conducta, los cuales pueden aplicarse a las variadas conductas aprendidas durante el proceso educativo durante la infancia.

El ambiente en el que el niño vive determina de esta manera su comportamiento y en muchas de las ocasiones las circunstancias que lo rodean le obligan a desplegar conductas negativas, destructivas para sí mismo o para el grupo, y los padres, aunque desconociendo las causas y motivos, tienen que enfrentarse a esas conductas y manejarlas, cumpliendo así con su tarea de autoridad educativa que la sociedad le impone tradicionalmente; y lógicamente la respuesta de los padres a gran parte del comportamiento del niño se da por medio de castigos, que van desde la simple desaprobación hasta acciones físicas como golpes y aislamiento y que desde luego aumentan las situaciones conflictivas que el niño tiene que vivir y que a su vez generan más conductas inadecuadas y se inicia un nuevo círculo vicioso.

Pero en todo el proceso de adquisición conductual, juega un papel muy importante el contexto socio-cultural en el que la familia se desenvuelve, ya que como regla general, los padres tienen como única guía patrones o pautas de conducta socio-culturalmente establecidos en relación a su interacción con los hijos, y muy pocos de ellos tienen la posibilidad de modificar por sí solos relaciones y actitudes incluidas en tales patrones que en cierta medida se han robado y no facilitan, sino dificultan el aprendizaje infantil.

4.1.3. Objetivo general

El Programa de Modificación de Conducta, se plantea como objetivo general proporcionar servicio a la comunidad a través de la aplicación de principios conductuales, a fin de dar dirección a conductas que interfieren con el desarrollo psicoeducativo del niño; generando a partir de esta investigaciones que permitan cambios en la evaluación, diagnóstico y tratamiento a la población atendida, así como proyectar el servicio derivado de la vinculación con la investigación, lo anterior, con la participación de estudiantes de licenciaturas de carreras afines.

4.1.4. Población atendida

El programa está dirigido a niños que presenten desórdenes dentro de su repertorio conductual; entendiéndose como desorden conductual, la emisión de conductas adecuadas con baja emisión y/o de las conductas inadecuadas del alta o media emisión, que impiden el desarrollo integral eficaz del niño.

4.1.5. Metodología

Dado que el objetivo del programa es el manejo conductual, se realizarán actividades académicas específicas por medio de las cuales se ejerza un manejo adecuado de las conductas que el niño presenta; sin pretender con esto restarle importancia a nuestro objetivo y tomando estas actividades sólo como apoyo para el trabajo conductual (Herramientas); estas actividades se integran dentro del Sub-Programa Nociones Básicas, el cual está conformado por los siguientes temas, entre otros:

- **ESQUEMA CORPORAL.** Cuyo objetivo es identificar y nombrar las partes y funciones del cuerpo humano.
- **COLORES.** Su objetivo, es identificar, discriminar y manejar adecuadamente los colores.
- **MOTRICIDAD FINA.** Adquirir y/o mejorar la coordinación motora del niño.
- **FIGURAS GEOMÉTRICAS.** Identificar, discriminar figuras geométricas.
- **NOCIONES TEMPORALES.** Identificar, discriminar y manejar adecuadamente su tiempo y su espacio.
- **PERCEPCIÓN VISUAL.** Interpretar adecuadamente los estímulos visuales.

4.1.6. Implementación de la biblioteca en caso de estudio

El diagrama de clases correspondiente a la implementación de la biblioteca en un caso de estudio se muestra en la Figura 4.1, en donde además de muestra la implementación en el patrón arquitectónico Modelo Vista Controlador.

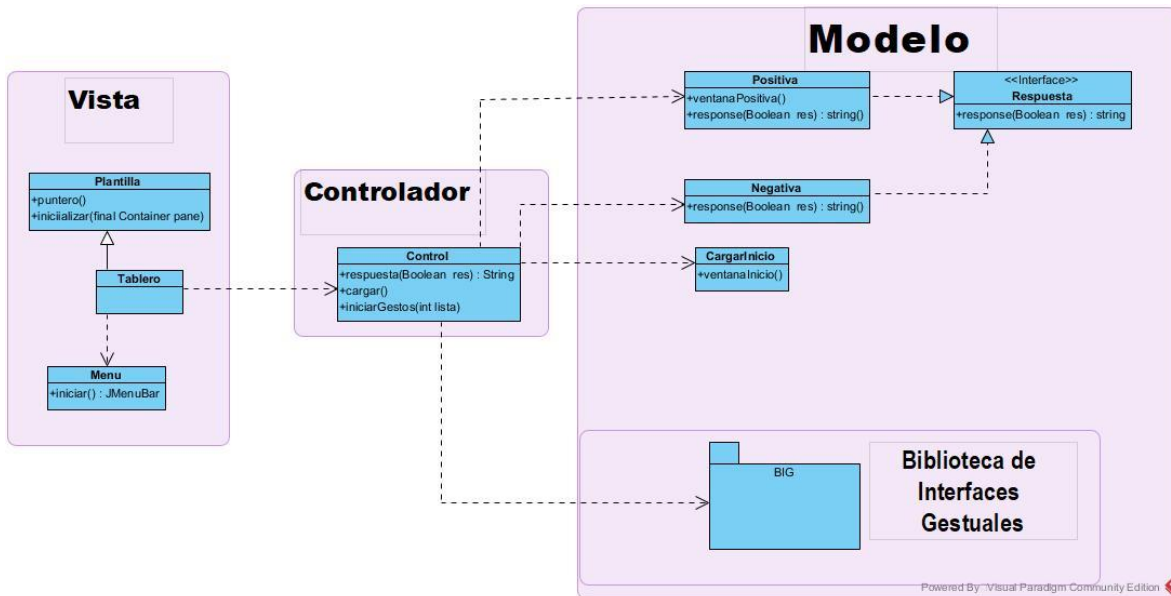


Figura 4. 1. Diagrama de Clases de la implementación de la biblioteca de interfaces gestuales en caso de estudio.

El sistema cuenta con el uso de tableros los cuales implementarán las actividades descritas en el punto 4.1.5. Metodología, estos tableros serán manejados mediante la implementación de la biblioteca de interfaces gestuales como apuntador, el cual será manejado mediante el uso de los gestos faciales, de acuerdo a los requerimientos de algunos niños será como se configure la aplicación.

El sistema iniciará con una pantalla de bienvenida como se muestra en la Figura 4.2 en la cual el alumno ya podrá hacer uso de la biblioteca de interfaces gestuales. Para iniciar las actividades se tiene que dar clic en el botón de iniciar como indica la Figura 4.3.



Figura 4. 2. Pantalla de Inicio.



Figura 4. 3. Pantalla de Inicio.

Al iniciar los tableros se activa una grabación que dará la instrucción a seguir, las cuales son que se seleccione la figura adecuada para esa actividad, si la figura seleccionada no es la correcta, en la parte inferior se mostrará un mensaje en el cual se indicará que la figura seleccionada es incorrecta, en cada implementación

del sistema las imágenes aparecerán en diferente orden así el alumno no podrá memorizar en donde se localiza cada imagen, las partes del cuerpo las cuales aparecen en el tablero son: cabeza, brazo, ojos, dedo, pie, oreja.

Se iniciará con las actividades de esquema corporal cuyo objetivo es identificar las partes del cuerpo, para cumplir este objetivo se cuenta con tres tableros los cuales contienen diferentes partes del cuerpo, el objetivo del tablero número uno es que el alumno identifique cuál es la cabeza como se muestra en la Figura 4.4, si el alumno selecciona una imagen diferente, aparecerá el error como se indica en la Figura 4.5, tal error es el mencionado anteriormente.

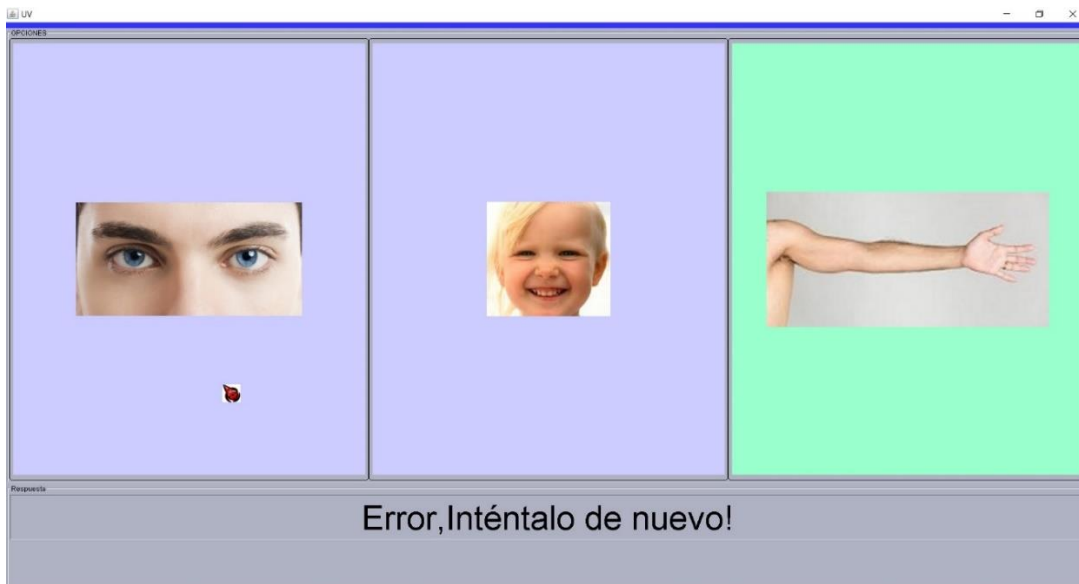


Figura 4. 4. Error en el tablero por no haber seleccionado la imagen de la cabeza.

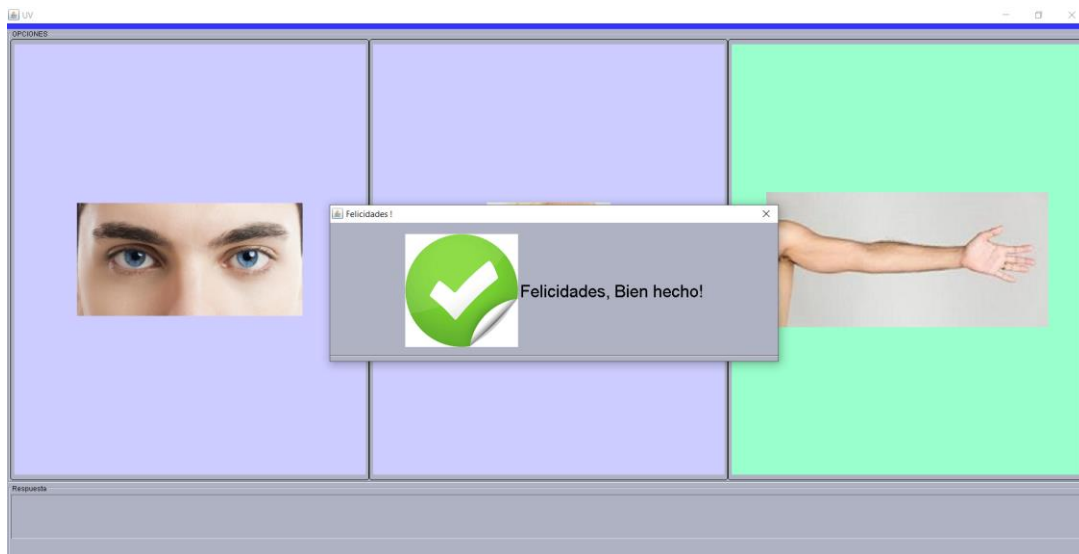


Figura 4. 5. Ventana cuando se selecciona la figura cabeza correctamente.

El tablero número dos contiene las mismas imágenes del cuerpo humano que el tablero número uno, pero con la diferencia que aparecen en diferente orden en el tablero, aquí su objetivo es identificar dónde se localiza el brazo, por otro lado, si se selecciona otra parte del cuerpo, se mostrará un error en la parte inferior del tablero y por lo consiguiente no permitirá avanzar al siguiente tablero como se muestra en la Figura 4.6, al seleccionar el brazo se avanzará al siguiente tablero tal y como se aprecia en la Figura 4.7.



Figura 4. 6. Error en el tablero por no haber seleccionado la figura del brazo.

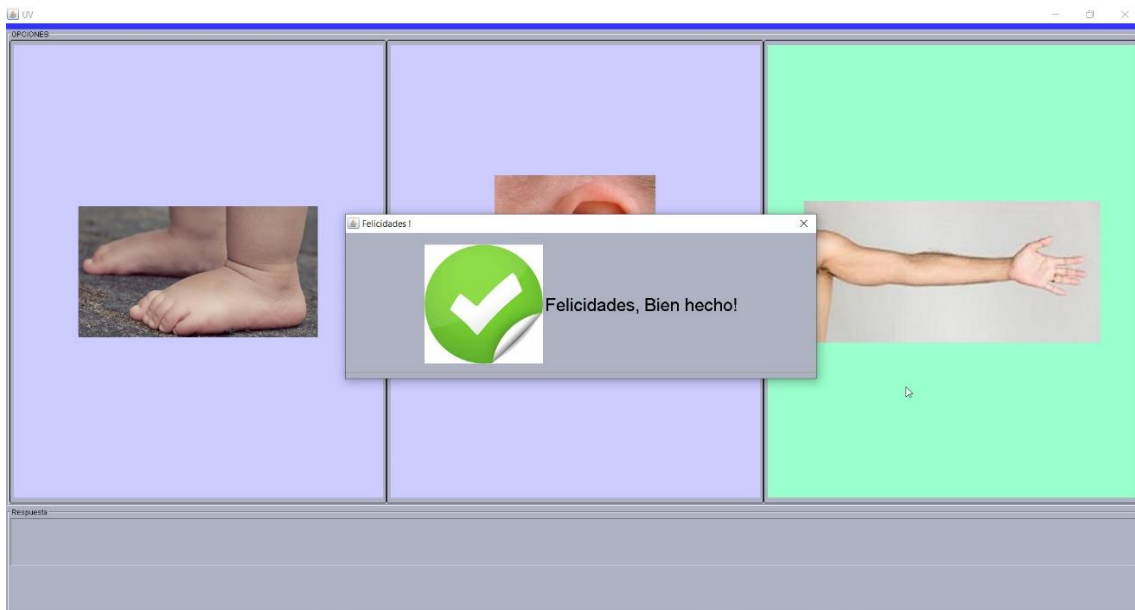


Figura 4. 7. Ventana cuando se selecciona la figura brazo correctamente.

Una vez avanzado al tablero número tres el alumno tendrá como objetivo identificar cuáles son los pies, para ello la grabación le indicará que los ubique y de clic sobre la figura de los pies para que cambie de actividad, si selecciona la figura que no corresponda a los pies mostrará el error como se muestra en la Figura 4.8, cuando se selecciona la figura correcta se pasa al siguiente tablero de actividades como se muestra en la Figura 4.9.

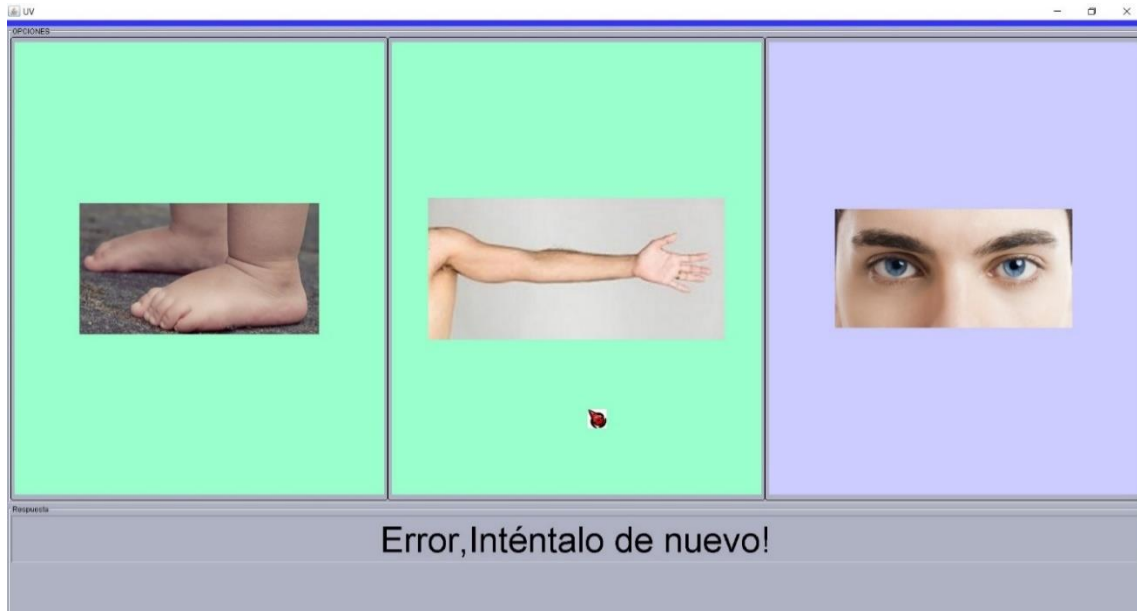


Figura 4. 8. Error en el tablero por no haber seleccionado la figura de los pies.

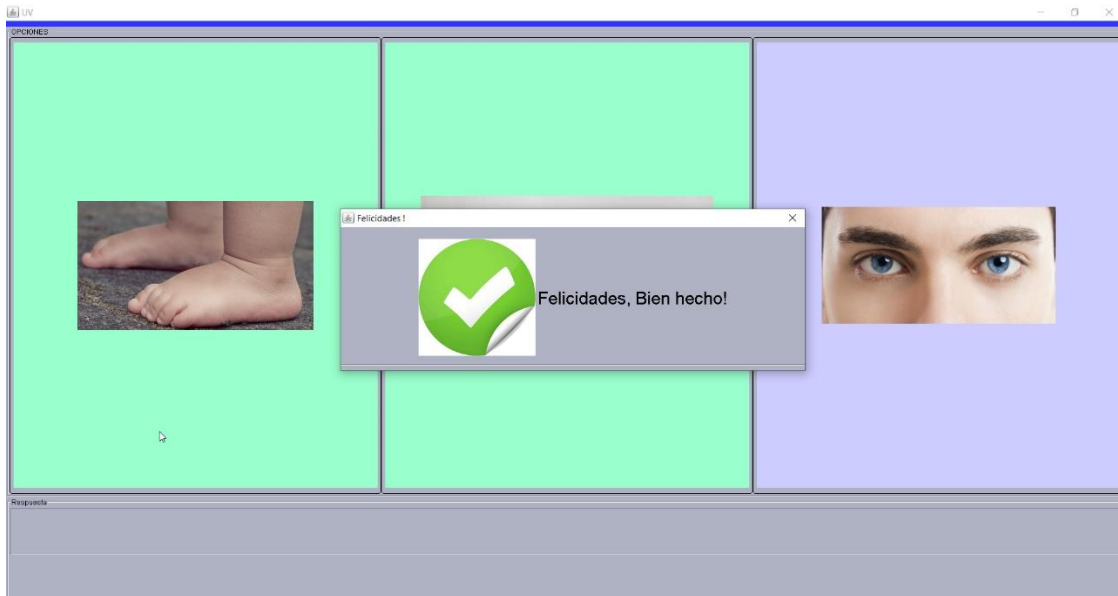


Figura 4. 9. Ventana al seleccionar la figura pies correctamente.

Una vez terminadas las actividades de esquema temporal se inician las actividades de nociones temporales con lo cual se busca que el alumno tenga como objetivo identificar, discriminar y manejar adecuadamente su tiempo y su espacio, para esto se crean cinco tableros los cuales contienen dichas actividades que se describen a continuación.

El tablero número cuatro contiene la actividad que solicita que el alumno identifique cuál es el lápiz más grande, si selecciona el lápiz pequeño, el tablero mostrará error y no cambiará de actividad como se ve en la Figura 4.10, al seleccionar el más grande cambiará a la siguiente actividad como se indica en la Figura 4.11.

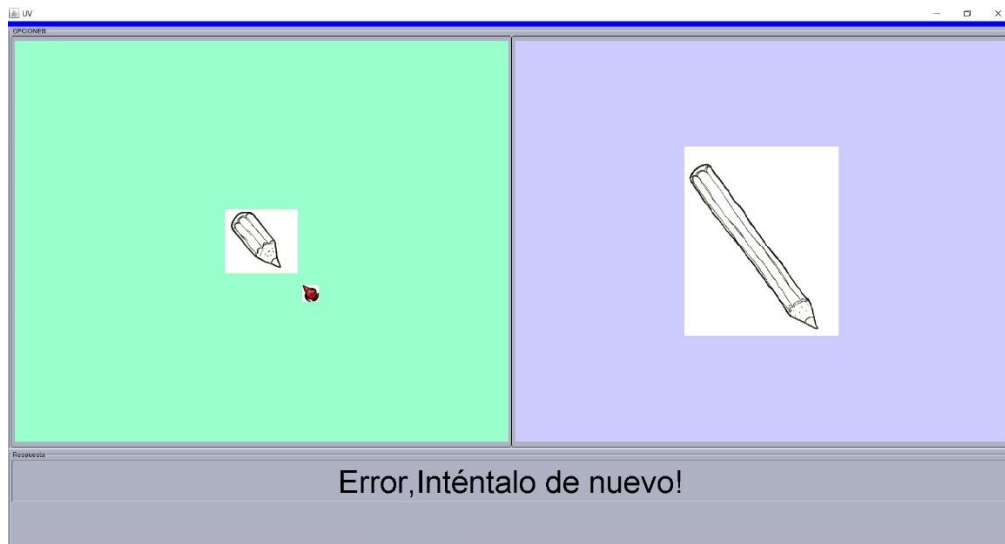


Figura 4. 10. Error en el tablero al haber seleccionado el lápiz pequeño.

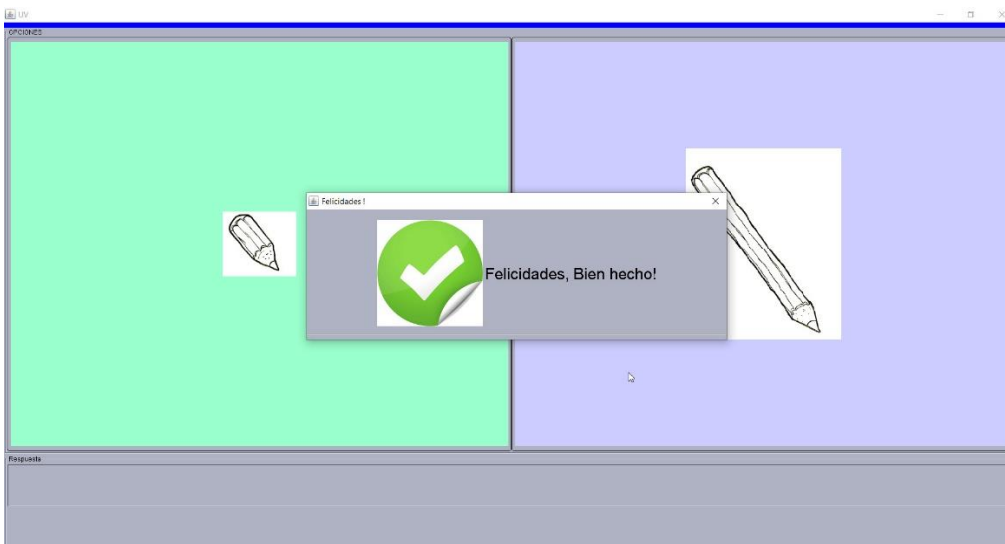


Figura 4. 11. Ventana al seleccionar el lápiz más grande.

El tablero número cinco que corresponde a la actividad de nociones temporales tiene como objetivo que el alumno identifique dónde es arriba y dónde es abajo, por lo cual debe de identificar donde el muñeco esté debajo de la mesa, si elige el equivocado, el tablero mostrará el error como lo muestra la Figura 4.12 y no cambiará de tablero, si el alumno acierta, cambia de tablero como se observa en la Figura 4.13.

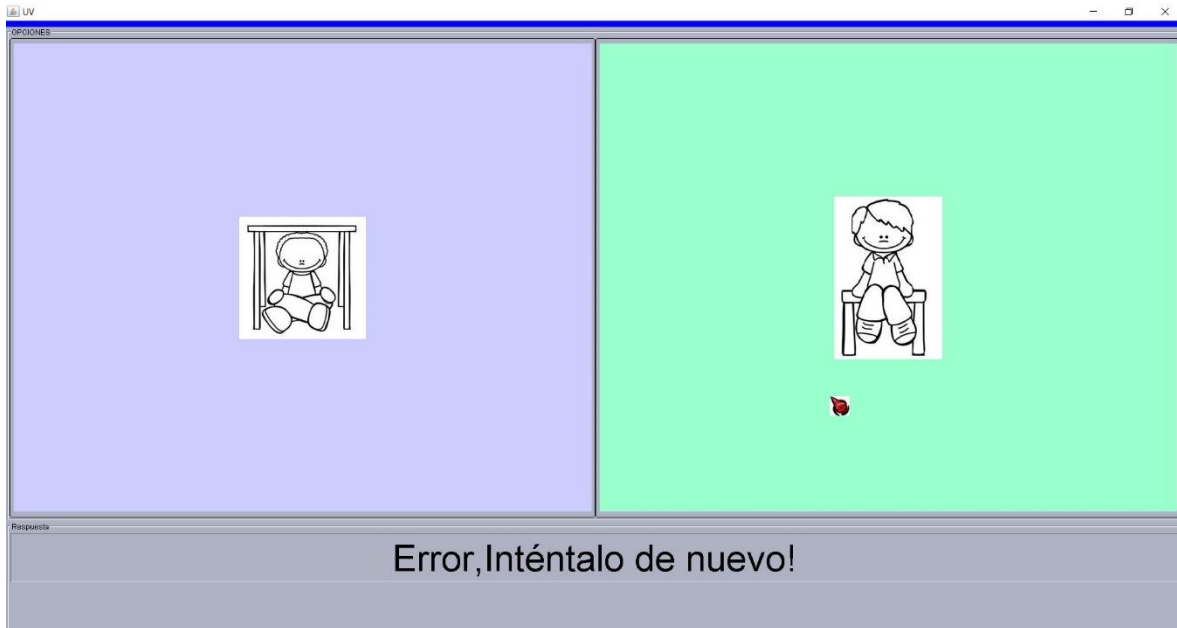


Figura 4. 12. Error en el tablero al haber seleccionado imagen incorrecta.

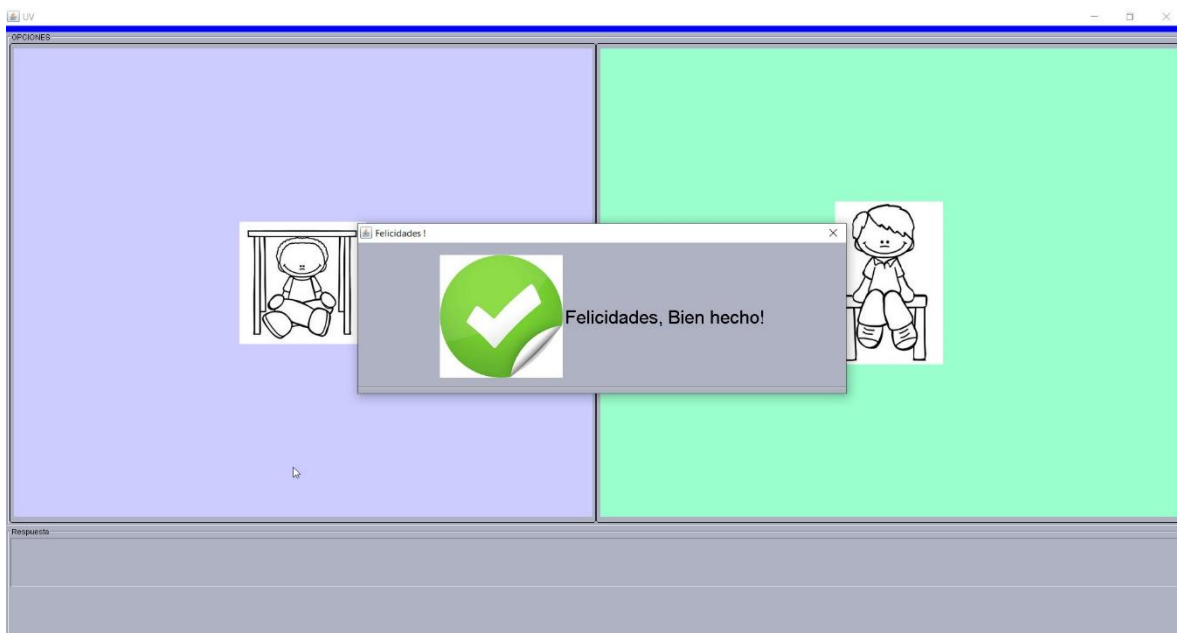


Figura 4. 13. Ventana que aparece al haber elegido la figura correcta.

La tercera actividad correspondiente a nociones temporales se muestra en el tablero número seis el cual corresponde al objetivo que el alumno identifique el muchos y pocos, en el tablero se muestran dos figuras las cuales una contiene muchos lápices y en la otras menos, debe seleccionar la imagen que contiene más lápices, si elige la que contiene menos, mandará en la parte inferior del tablero un error y no dejará avanzar a la siguiente actividad como se muestra en la Figura 4.14, al seleccionar la correcta presentará una ventana la cual mostrará al alumno que ha elegido la correcta y avanzará de actividad al siguiente tablero como se observa en la Figura 4.15.

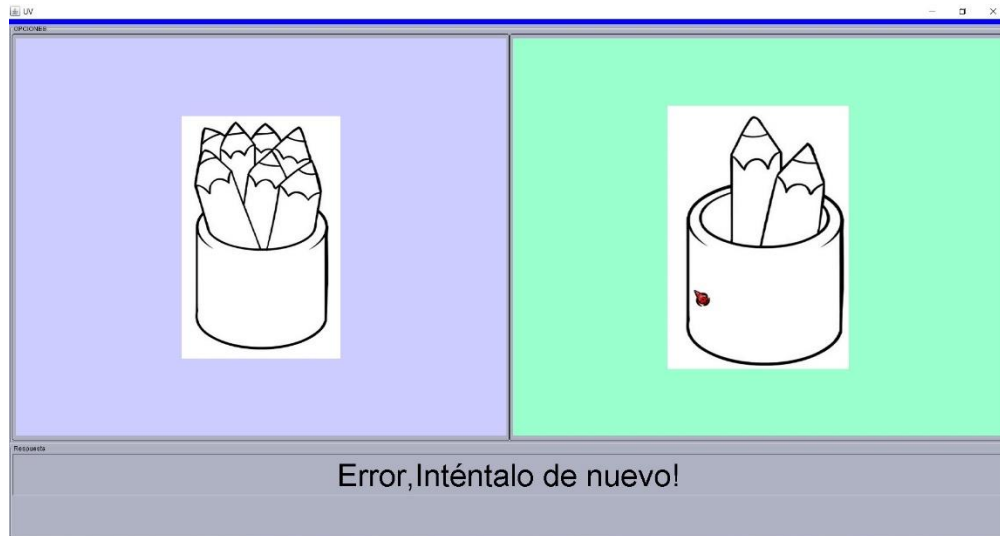


Figura 4. 14. Error en el tablero al haber elegido imagen con menos lápices.

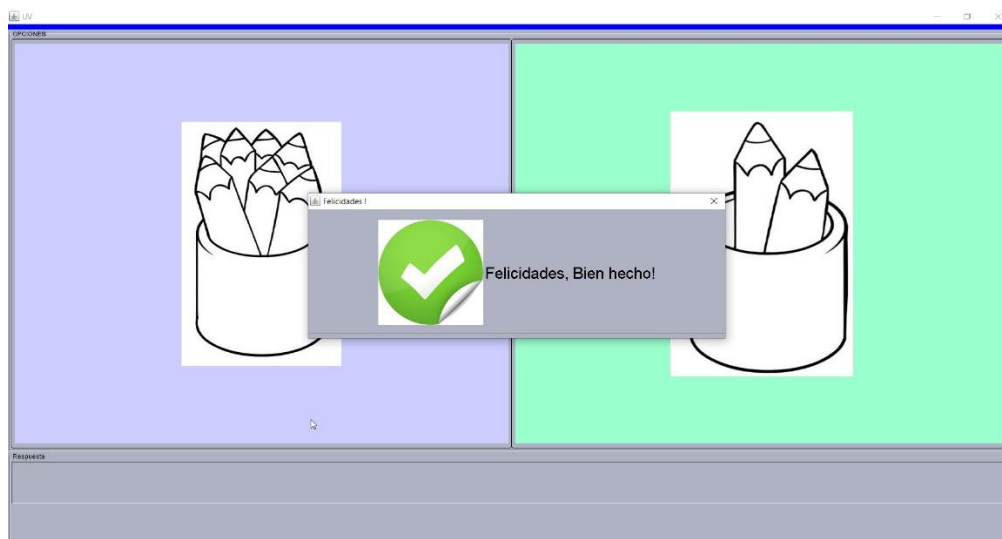


Figura 4. 15. Ventana al haber seleccionado la figura que contiene más lápices.

Continuando con las actividades sobre nociones temporales se llega al tablero número siete después de haber acertado en la opción correcta en el tablero número seis, en esta actividad se tiene como objetivo que se indique cuál figura es la que contiene la imagen donde es de día, si el alumno elige la imagen donde es de noche el tablero mostrará un error y no permitirá continuar con la siguiente actividad como se muestra en la Figura 4.16, cuando se elige la figura correcta la cual en esta actividad es la que muestra que es de día, permitirá al alumno avanzar al siguiente tablero como se observa en la Figura 4.17.

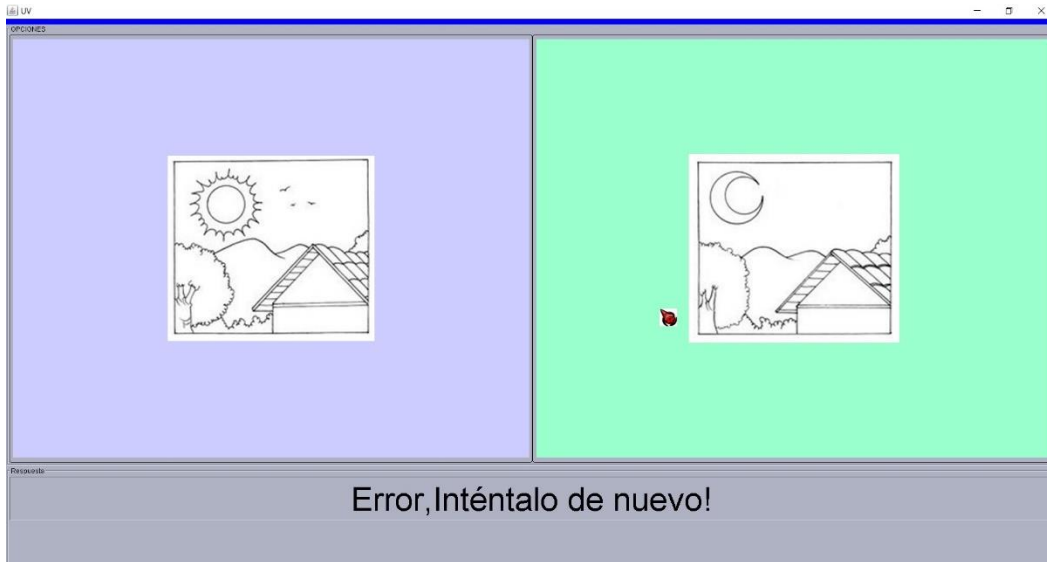


Figura 4. 16. Error en el tablero al haber elegido figura donde es de noche.



Figura 4. 17. Ventana al haber elegido la figura donde es de día.

Al acertar en el tablero número siete se llega al fin de las actividades acerca de nociones temporales, en esta última actividad se busca que el alumno comprenda qué se debe hacer en el día y qué se debe hacer en la noche, por lo cual la grabación pedirá al alumno elija la imagen sobre qué se debe de hacer en la noche, lo cual es dormir, si elige la figura donde un niño está jugando, el tablero mostrará error y no dejará avanzar al alumno de actividad como se observa en la Figura 4.18, cuando el alumno seleccione correctamente la figura donde el niño duerme avanzará a la actividad del siguiente tablero como se muestra en la Figura 4.19.

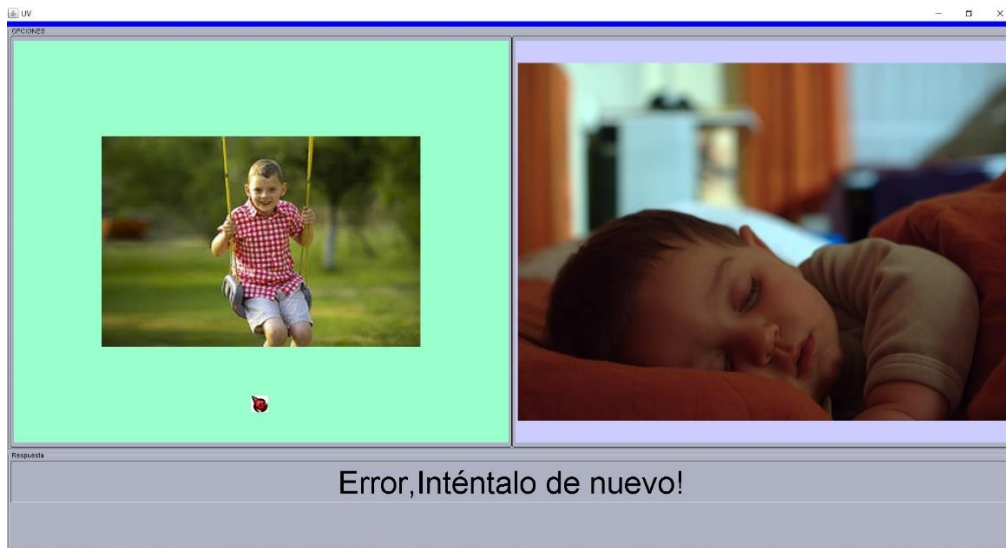


Figura 4. 18. Error en el tablero al haber elegido figura donde es de noche.

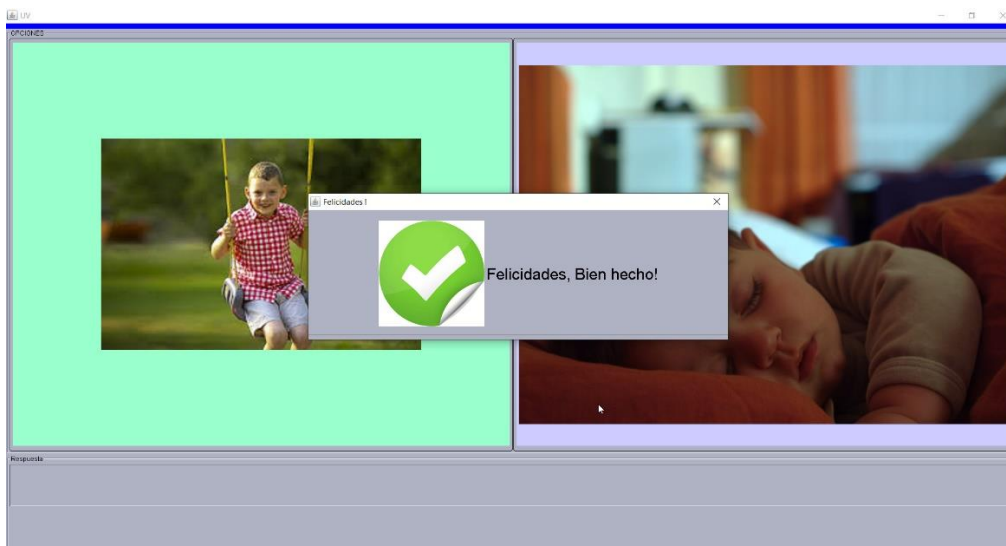


Figura 4. 19. Ventana al haber elegido la figura correcta donde el niño duerme.

Terminando las actividades en los tableros acerca de nociones temporales, se inicia con las actividades de colores que tiene como objetivo que el alumno identifique y maneje adecuadamente los colores.

Esta actividad cuenta con dos tableros en los cuales se busca que identifiquen diferentes colores, en el tablero número nueve se pide al alumno que identifique el color verde como se muestra en la Figura 4.20, en el tablero número diez se le pide al alumno que seleccione el color amarillo como se observa en la Figura 4.21. En las dos actividades si el alumno elige un color diferente al que se le pide que indique no lo dejará avanzar de actividad y se le mostrará error en la parte inferior del tablero.

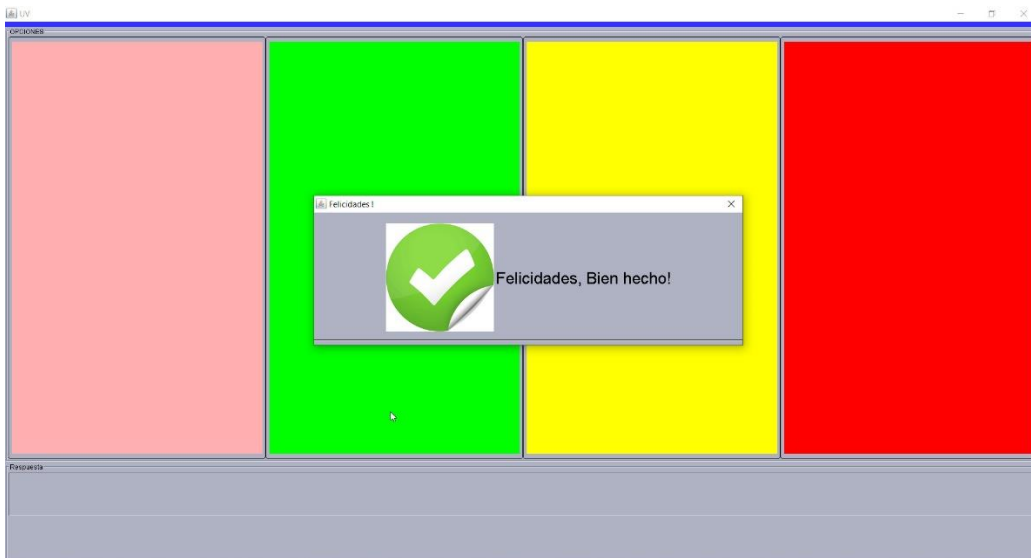


Figura 4. 20. Actividad para identificar el color verde.

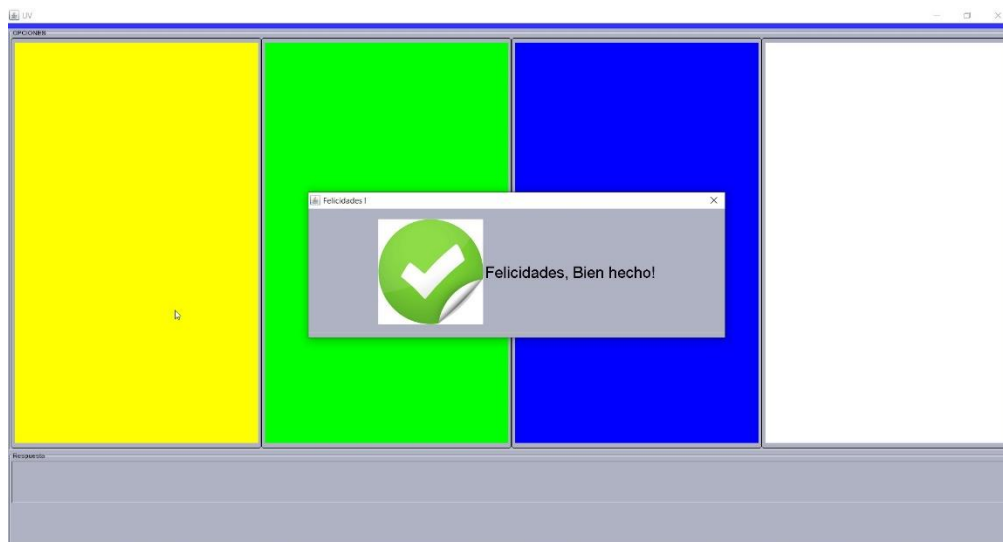


Figura 4. 21. Actividad para identificar el color amarillo.

Finalizando con los objetivos de identificación de colores en los dos tableros anteriores, se prosigue con las dos actividades las cuales sus tableros tienen el objetivo que el alumno identifique y discrimine figuras geométricas, en los tableros se muestra una figura en la parte superior en donde el alumno debe identificar cuál de las tres imágenes de abajo es la que es igual a ésta. Como en actividades pasadas si un alumno elige una figura la cual es diferente a la mostrada en la parte inferior, mostrará una alerta de error y solo avanzará de actividad si elige la figura correcta, en las Figuras 4.22 y Figura 4.23 se muestran las actividades con las actividades correspondientes a figuras geométricas.

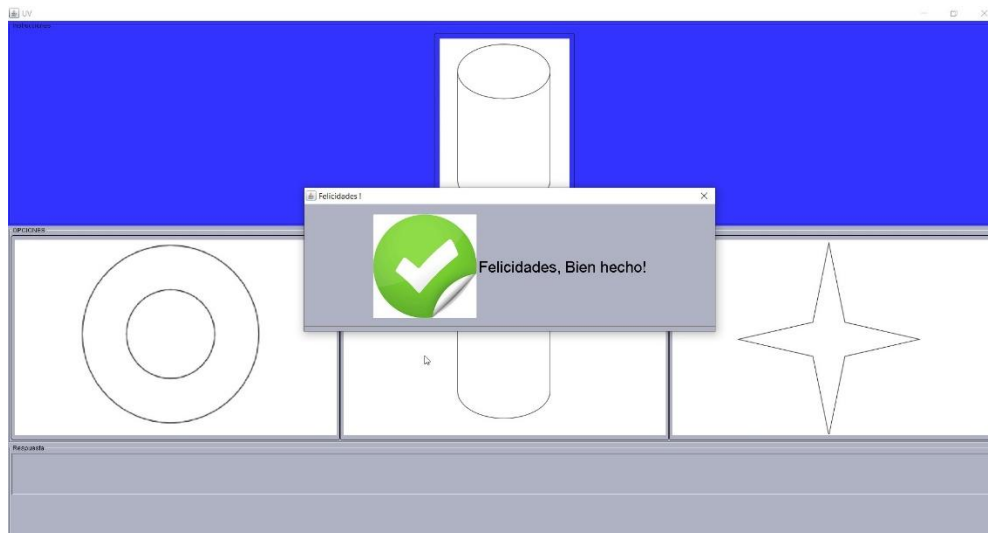


Figura 4. 22. Actividad de identificar un cilindro.

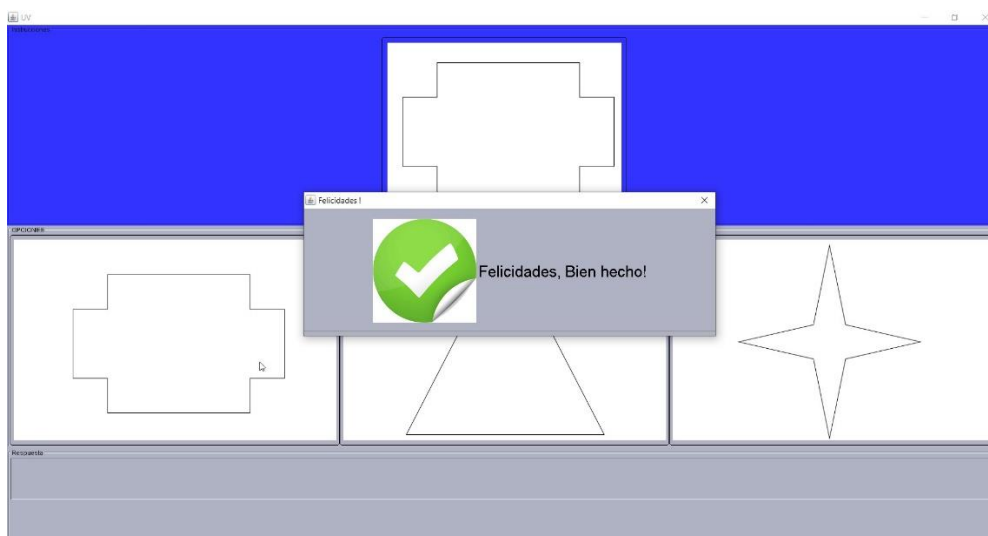


Figura 4. 23. Actividad de identificar una cruz.

Al haber finalizado las actividades de figuras geométricas se prosigue con las actividades de percepción visual, en donde el alumno tiene como objetivo interpretar adecuadamente los estímulos visuales, por ello en los tableros se mostrarán tres figuras en las cuales el alumno debe identificar la figura que es diferente a las demás, como en las actividades pasadas si el alumno elige una opción incorrecta se mostrará un error en forma de alerta en la parte inferior, seleccionando la figura correcta se avanza de nivel, en la Figura 4.24 y Figura 4.25 se muestran las imágenes de las actividades correspondientes.



Figura 4. 24. Actividad de identificar una cara al revés.



Figura 4. 25. Actividad de identificar una luna al revés.

Con las acciones anteriores se concluyen las actividades del caso de estudio, por lo que prosigue las implementaciones de la biblioteca en el caso de estudio, para lo cual primeramente se hace un análisis en los niños seleccionados para el uso del caso de estudio, su participación es dada por la Lic. Laura Patricia Medrano Herrera.

4.3 Características de la implementación del caso de estudio

Para la implementación del caso de estudio se contó con la participación de 10 niños que asisten a la Universidad Veracruzana/Instituto de Psicología y Educación, Centro de Entrenamiento y Educación Especial, la población de niños a los que se les aplicó la implementación del caso de estudio cuenta con las siguientes características:

- 10 alumnos:
 - 9 alumnos de sexo masculino.
 - 1 alumno de sexo femenino.
- El rango de edad es de 5 a 11 años.
- Con todo alumno se ensayó cada seña de la biblioteca
- Tienen diferente tiempo en la institución de 0 a 12 meses.

La configuración que se utilizó durante la aplicación del caso de estudio fue:

- Computadora Laptop Dell Inspiron 15 5000 Series.
- Procesador Core i7-8550U CPU @ 1.8GHZ 1.99 GHZ.
- 12 GB de memoria RAM.
- Windows 10 de 64 bits.
- Diadema Emotiv EPOC +.
- SDK Emotiv.
- Biblioteca de Interfaces Gestuales BIG.
- Solución salina Renu.

En la Figura 4.26 se aprecia el uso de la diadema con un alumno al cual se le implementó el caso de estudio.



Figura 4. 26. Dispositivos para el manejo de biblioteca de interfaces gestuales.

La implementación del caso de estudio se realizó bajo las siguientes características:

- La aplicación se realizó en diversos días en alrededor de dos semanas.
- Se trabajó en cubículos que se encuentran dentro en parte posterior del aula de la Lic. Laura Patricia Medrano Herrera.
- La Lic. Laura Patricia Medrano Herrera otorgó el consentimiento de fotografiar a los alumnos durante la implementación.
- La Lic. Laura Patricia Medrano Herrera dio el orden de cómo pasaban los alumnos a la implementación.
- La implementación para cada niño junto con el entrenamiento tuvo una duración de 30 a 40 minutos.
- En los cubículos se escuchaba ruido externo.

4.4 Resultado de Implementación de biblioteca para el manejo de interfaces humano-computadora a través de gestos faciales en un Sistema de modificación de conducta con niños de déficit de atención con o sin hiperactividad

Primeramente, cuando el niño era elegido por la Lic. Laura Patricia Medrano Herrera, se le aplicaba el cuestionario previo el cual consiste de cuatro preguntas, las cuales eran:

1. ¿Utiliza computadora para aprender?
 - a. sí
 - b. no
2. ¿Utiliza computadora para jugar?
 - a. sí
 - b. no

3. ¿Ha tenido experiencia previa con dispositivos de control gestual?
 - a. sí
 - b. no
4. Si la respuesta es sí, mencione cuál(es)

Tabla 4. 2. Resultado de cuestionario Previo.

Pregunta/No. Alumnos	¿Utiliza computadora para aprender?		¿Utiliza computadora para jugar?		¿Ha tenido experiencia previa con dispositivos de control gestual?	
	Sí	No	Sí	No	Sí	No
10	2	8	6	4	0	

A la licenciada se le aplicó un cuestionario previo el cual consistía en las siguientes preguntas:

1. ¿Utiliza computadora para realizar las actividades de enseñanza?
 - a. sí
 - b. no
2. ¿Utiliza computadora para preparar materiales educativos?
 - a. sí
 - b. no
3. ¿Ha tenido experiencia previa con dispositivos de control gestual?
 - a. sí
 - b. no
4. Si la respuesta es sí. Mencione cuál(es)

Después del cuestionario previo mostrado en Tabla 4.2, a cada niño se le explicó en qué consistía el caso de estudio, qué es lo que debía realizar de modo que se le hiciera atractivo. Por la cuestión de tiempo y de usabilidad de la diadema, antes de que cada niño iniciará su participación, los electrodos de la diadema eran bañados con solución salina RENU, posteriormente se realizaba el entrenamiento de gestos faciales con ayuda de EmotivBCI, cada entrenamiento facial era ensayado diez veces para que el entrenamiento fuera de la mejor manera, en la Figura 4.27 se muestra el entrenamiento de dos alumnos.

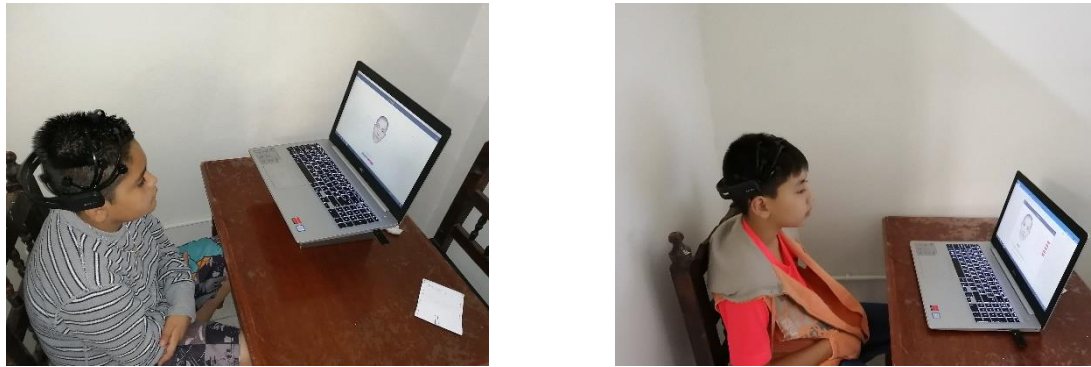


Figura 4. 27. Entrenamiento Gestual.

Posterior al entrenamiento se eligieron los gestos que realizaría cada alumno dependiendo de sus capacidades físicas, dado que algunos alumnos no pueden cerrar solamente un ojo, deben cerrar los dos, algunos alumnos si podían cerrar el ojo derecho, pero no el ojo izquierdo y viceversa; algunos podían sonreír, pero no apretar los dientes, algunos su sonrisa parecía más que apretaban los dientes; algunos se les dificultaba el seguimiento ocular, en la tabla 4.3 se muestran los diferentes gestos faciales que son parte de la biblioteca y los cuales se utilizaron en la implementaron en el caso de estudio, en la tabla 4.4 se muestran los comandos que son utilizados

Tabla 4. 3. Gestos Faciales implementados en caso de estudio.

Nombre	Gesto
Left wink (Guiño Izquierdo)	
Right Wink (Guiño Derecho)	
Smile (Sonreír)	
Looking Left (Mirada izquierda)	

Tabla 4. 4. Gestos Faciales implementados en caso de estudio.

Nombre	Gesto
Wink (Guiño)	
Clench (Apretar)	
Looking Right (Mirada Derecha)	

Tabla 4. 5. Comandos implementados en el caso de estudio.



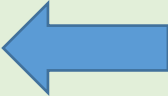

Nombre	Acción
Arriba	
Abajo	
Izquierda	
Derecha	

Tabla 4. 6. Comandos implementados en el caso de estudio.

Nombre	Acción
Clic Izquierdo	

Por los impedimentos físicos de algunos alumnos se tuvieron diferentes implementaciones de la biblioteca de gestos faciales en el caso de estudio, en la tabla 4.5 las implementaciones que fueron necesarias durante la implementación de la biblioteca de gestos faciales en el caso de estudio.

Tabla 4. 7. Comandos implementados en el caso de estudio.






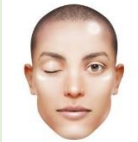


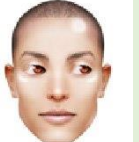
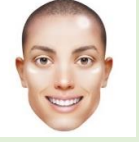







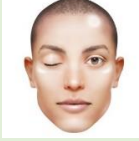
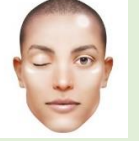
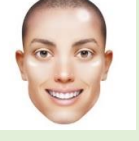
Implementación	 Derecha	 Izquierda	 Arriba	 Abajo	 Clic
1	 Guiño Derecho	 Guiño Izquierdo.	 Mirada Derecha.	 Mirada Izquierda.	 Sonrisa.
2	 Mirada Derecha.	 Mirada Izquierda.	 Guiño Derecho	 Guiño Izquierdo.	 Sonrisa.
3	 Mirada Derecha.	 Mirada Izquierda.	 Guiño Derecho	 Guiño Izquierdo.	 Sonrisa.

Tabla 4. 7. Comandos implementados en el caso de estudio.

































Implementación	 Derecha	 Izquierda	 Arriba	 Abajo	 Clic
4	 Guiño Derecho	 Guiño Izquierdo.	 Mirada Derecha.	 Mirada Izquierda.	 Sonrisa.
5	 Guiño Derecho	 Guiño Izquierdo.	 Mirada Derecha.	 Mirada Izquierda.	 Sonrisa.
6	 Sonrisa.	 Guiño			
7	 Guiño	 Sonrisa.	 Guiño Derecho	 Guiño Izquierdo.	 Apretar.
8	 Sonrisa.	 Guiño Derecho	 Apretar.	 Mirada Izquierda.	 Guiño
9	 Guiño Izquierdo.	 Apretar.	 Mirada Derecha.	 Mirada Izquierda.	 Sonrisa.

Tabla 4. 7. Comandos implementados en el caso de estudio.





Implementación	 Derecha	 Izquierda	 Arriba	 Abajo	 Clic
10	 Guiño Derecho	 Guiño Izquierdo.	 Mirada Derecha.	 Mirada Izquierda.	 Sonrisa.



Figura 4. 28. Participación en caso de estudio.

Al término de la aplicación a cada a alumno se realizó un pequeño cuestionario para la retroalimentación del caso de estudio.

Cuestionario posterior del Alumno.

1. ¿La aplicación te parece divertida?
 - a. Sí
 - b. no
2. ¿Usar la diadema te parece incómodo?
 - a. Sí
 - b. no
3. Menciona qué te gustó de la aplicación.
4. Menciona que no te gustó de la aplicación.
5. ¿Alguna sugerencia?

Tabla 4. 8. Resultado de cuestionario posterior.

Pregunta/ No. Alumnos	La aplicación te parece divertida		Usar la diadema te parece incómodo.	
	Si	No	Si	No
10	9	1	3	7

Los resultados del cuestionario posterior mostrados en la Tabla 4.6 se observa que en la pregunta 3 la mayoría de los alumnos respondieron que les gusto el apuntador y el interactuar con la computadora, así como las imágenes de las actividades. En la pregunta 4 respondió solo un alumno que no le gustó la interacción de los gestos con la computadora, los demás alumnos no respondieron esa pregunta porque les resultó satisfactorio el sistema. En la pregunta 5 todos respondieron que quisieran más aplicaciones.

La licenciada también tuvo un cuestionario posterior a las aplicaciones:

Cuestionario posterior del profesor

1. ¿El uso del control gestual aumenta la atención del usuario?
 - a. Sí
 - b. no
2. ¿El uso de la aplicación genera cansancio o incomodidad?
 - a. Sí
 - b. no
 - c. Algunas veces.
3. ¿Las opciones que se presentan son relevantes y oportunas?
 - a. Sí
 - b. no
4. ¿La aplicación es útil para identificar las necesidades de los usuarios?
 - a. Sí
 - b. no
5. ¿La aplicación mantiene la atención del usuario?
 - a. Sí
 - b. no
6. ¿El uso de las imágenes es correcto?
 - a. Sí
 - b. no
7. ¿El tiempo de respuesta de la aplicación es adecuado?
 - a. Sí
 - b. no
 - c. Algunas veces.
8. Mencione los aspectos positivos de la aplicación.

9. Mencione los principales problemas de la aplicación.

10. Otros comentarios o sugerencias para mejorar la aplicación.

Tabla 4. 9. Resultado de cuestionario Posterior del Profesor.

Pregunta/Respuesta	Sí	No	Algunas veces
1 ¿ El uso del control gestual aumenta la atención del usuario?	X		
2 ¿ El uso de la aplicación genera cansancio o incomodidad?			X
3 ¿Las opciones que se presentan son relevantes y oportunas?	X		
4 ¿La aplicación es útil para identificar las necesidades de los usuarios?	X		
5 ¿La aplicación mantiene la atención del usuario?	X		
6 ¿El uso de las imágenes es correcto?	X		
7 ¿El tiempo de respuesta de la aplicación es adecuado?	X		
8 Mencione los aspectos positivos de la aplicación: 1. Mantiene la atención del usuario 2. Le es atractivo por el manejo de la computadora. 3. Apoya en el establecimiento y manejo de conceptos. 4. Desarrolla el aprendizaje			
9 Mencione los principales problemas de la aplicación: 1. A algunos niños les resultó incómoda la diadema. 2. Los conceptos no fueron atractivos para el niño más grande. 3. Las instrucciones para realizar los gestos, algunos de los niños no sabían hacerlos.			
10 Otros comentarios o sugerencias para mejorar la aplicación: La aplicación es muy buena pues como ya dije, permite estimular la consolidación de repertorio básico conductual; de igual manera les permite a los niños desarrollar el establecimiento y manejo de conceptos para su aprendizaje. Los puntos que considero deberían considerarse para la mejora de la aplicación son: 1. Para los niños pequeños gesticular se les complica y eso pudiera influir en el uso de la diadema. 2. Temas (Español, Matemáticas) para niños de mayor edad(9-11 años).			

Con esto se terminó la aplicación del caso de estudio para dar paso a la obtención de resultados.

Capítulo 5. Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones al finalizar el proyecto tesis y con ello se dan recomendaciones para trabajar en un futuro, dichas recomendaciones se obtuvieron de la implementación del caso de estudio.

5.1. Conclusiones

En el cuestionario previo que se muestra en la página 78 aplicado a cada alumno, se determinó en la Tabla 4.2 que de los 10 alumnos a los cuales se les aplicó el caso de estudio, ninguno tenía experiencia previa con dispositivo de control gestual, por lo cual la interacción que iban a realizar era algo nuevo y novedoso para ellos. Solo dos de los alumnos ocupan la computadora para tareas y seis alumnos usan la computadora para jugar y cuatro no han ocupado la computadora.

Con el cuestionario realizado previamente a la aplicación a la Licenciada se logró determinar que utiliza la computadora para realizar sus actividades de enseñanza y preparar sus materiales educativos, pero que no cuenta con la experiencia con dispositivos de control gestual, los resultados del cuestionario posterior se muestran en la Tabla 4.7.

Como se observa en la Tabla 4.5 hubo diversos tipos de implementaciones en el caso de estudio, dado que los niños tienen diferentes capacidades físicas en el manejo de gestos faciales, por lo cual se demostró que dado sus diversas implementaciones de la biblioteca de interfaces gestuales en un sistema de Java Swing cumple con su objetivo principal, el cual es tener una herramienta que ayude a los desarrolladores a tener diversas opciones en una biblioteca para el manejo de interfaces humano-computadora a través de gestos faciales, con lo cual le da diversas opciones de gestos los cuales se ocupan dependiendo de las necesidades que se tenga.

Se logra concluir que personas con impedimento físico es complicado que usen la diadema Emotiv Epoc + dado que sus limitaciones físicas no ayudan al manejo de los gestos faciales ya que no son lo necesariamente pronunciados para que la diadema los reconozca.

Durante la implementación de pruebas del caso de estudio con los alumnos se encontraron una serie de problemas con el uso de la diadema, ya que al hacer algunas veces algún gesto se mueven partes de la cara sin querer y eso hace que se la diadema reconozca un gesto que los alumnos no querían, por lo cual en la siguiente sección se indican posibles correcciones de mejoras las cuales se encontraron durante las estancias realizadas y las cuales se pudieran implementar

en un futuro para una mayor eficacia en su funcionamiento y así solucionar estos detalles.

En las pruebas realizadas en la aplicación del caso de estudio se demostró que la diadema cuenta con un desfase de tiempo durante el envío de señales a la computadora, lo cual algunas veces hace que la ejecución del apuntador se desfase.

Aunque ya existen investigaciones realizadas a la diadema Emotiv Epoc + no se encontró ninguno que haya creado una biblioteca para el manejo de interfaces humano-computadora a través de gestos faciales, así que esta tesis permite validar la eficacia y funcionalidad de la biblioteca implementándola de diversas maneras en un sistema de Java Swing bajo la arquitectura propuesta que es el patrón arquitectónico Modelo-Vista-Controlador todo esto recibiendo las señales cerebrales por medio de la diadema Emotiv Epoc +.

Con las pruebas realizadas en la implementación de la biblioteca en usuarios finales ayudó a conocer y a identificar comportamientos por parte de ellos, obteniendo una retroalimentación que sirve de punto de partida para recomendaciones propuestas que se detallarán en la sección posterior y con lo cual futuramente se podrán obtener mejoras, tales como el desfase que existe en la obtención de señales con la diadema Emotiv Epoc + y movimientos involuntarios ejecutados por el usuario.

Al usar la diadema en el trabajo se encontró que con el paso del tiempo y el uso de la solución salina la diadema se tiende a poner frágil y se empieza a tener problemas con las piezas de los electrodos, las cuales se empezaron a desarmar y se tuvo que pegar con cinta adhesiva.

Con el cuestionario posterior a la implementación del caso de estudio se comprobó que el 90% de los usuarios finales (alumnos) tuvo una gran aceptación e interés de la biblioteca en el caso de estudio y a los cuales les gustaría contar con más casos de estudio los cuales contengan manejo de interfaces humano-computadora a través de gestos faciales, exactamente aplicaciones como videojuegos. También 70% de los usuarios dijeron que el uso de la diadema no fue lastimoso para ellos.

Con los resultados del cuestionario posterior a la aplicación hecho a la licenciada, determinó que el uso del control gestual aumenta la atención en el usuario y solo en algunos casos generó incomodidad, y en general todo fue positivo por lo que se requiere que haya más aplicaciones de este tipo para que los niños con problemas con déficit de atención e hiperactividad puedan interactuar con la computadora y así generar avances en sus tratamientos.

Al saber esto se determina que la biblioteca de interfaces gestuales tiene un gran alcance en el ámbito de la educación para niños con problemas conductuales.

5.2. Recomendaciones

Como producto de las pruebas realizadas del desarrollo de una biblioteca para el manejo de interfaces humano-computadora a través de gestos faciales en el caso de estudio seleccionado se consideran las siguientes recomendaciones para seguir la expansión de la biblioteca a futuro procurando una mejor implantación que la realizada anteriormente:

- Una mejora propuesta es la exploración de áreas de interés para la implementación de la biblioteca en otros casos de estudio.
- Se propone la combinación de gestos faciales para un comando.
- Una mejora es la ampliación del uso del apuntador en el uso del clic derecho, para obtener más características.
- Se propone como posible solución al desfase de la señal que existe entre la diadema y la computadora, realizar pruebas con técnicas de Machine Learnig.
- Se proponen el uso Machine Learnig para obtener señales de gestos faciales sin necesidad de algún dispositivo en el usuario.

Productos Académicos



Fernando Octavio Reynoso Martínez, María Antonieta Abud Figueroa, Silvestre Gustavo Peláez Camarena, Lisbeth Rodríguez Mazahua, Ulises Juárez Martínez.

PROTOTIPO DE INTERFAZ HUMANO-COMPUTADORA CONTROLADA A TRAVÉS DE GESTOS FACIALES

ReCIBE, Año 8 No. 1, octubre 2019.

Estado: Presentado.

REFERENCIAS

- [1] P Montuschi, A Sanna, F Lamberti, and G Paravati, "Human-Computer Interaction: Present and Future Trends," *Computing Now*, vol. 7, no. 9, September 2014, IEEE Computer Society [online]: <http://www.computer.org/web/computingnow/archive/september2014>.
- [2] "Cómo funciona el seguimiento ocular". Available: <https://www.tobiidynavox.com/es/sobre-tobii-dynavox/sobre-nosotros/como-funciona-seguimiento-ocular/>. [Accessed: 10/September/2018].
- [3] Parmar Dhiren, "HUMAN COMPUTER INTERACTION". Seminar Report. Indian Institute of Technology Kharaypur, Available: <http://sit.iitkgp.ernet.in/research/aut04seminar1/> . [Accessed: 14/September/2018].
- [4] Tech Target, "Natural user interface (NUI)". [online]. Available: <https://whatis.techtarget.com/definition/natural-user-interface-NUI>. [Accessed: 10/September/2018].
- [5] "Natural User Interfaces – What are they and how do you design user interfaces that feel natural?" [online]. Available: <https://www.interaction-design.org/literature/article/natural-user-interfaces-what-are-they-and-how-do-you-design-user-interfaces-that-feel-natural>. [Accessed: 10/September/2018].
- [6] "Comunicación verbal y no verbal, diferencias y bases". [online]. Available: <https://blog.grupo-pya.com/comunicacion-verbal-no-verbal-diferencias-bases/>. [Accessed: 10/September/2018].
- [7] "CANALES, UTILIDADES Y ÁREAS DE APLICACIÓN DEL LENGUAJE NO VERBAL". Available: <http://funwayresort.com/canales-utilidades-areas-aplicacion-del-lenguaje-no-verbal/>. [Accessed: 10/September/2018].
- [8] "LA EXPRESIÓN FACIAL DE LAS EMOCIONES: HISTORIA Y APLICACIONES". Available: <http://www.cienciacognitiva.org/?p=664>. [Accessed: 10/September/2018].
- [9] "EMOTIV EPOC+ 14 Channel Mobile EEG", [online]. Available: <https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeeg/>. [Accessed: 04/Octubre/2018].
- [10] "¿Qué es la tecnología Java y para qué la necesito?", [online]. Available: https://www.java.com/es/download/faq/whatis_java.xml. [Accessed: 04/Octubre/2018].
- [11] "Scrum & Agile", [online]. Available: <http://scrum.org.mx/scrum-agile/>. [Accessed: 04/Octubre/2018].

- [12] “The NetBeans Platform”, [online]. Available: <https://netbeans.org/features/>. [Accessed: 04/Octubre/2018].
- [13] A. Hawkes, A. Liu, R. Tashakkori, E. Jackson, J. Tate, S. Tashakkori, D. Kale, M. Parry ,” Using the Emotiv EPOC Neuroheadset as an EEGControlled BrainComputer Interface”, in Summer Ventures in Math and Science 2015, pp 1–15, June 2015.
- [14] G. Hornero, E. Font, J. Tejedo, O. Casas, “Interfaz gestual para el acceso a plataformas digitales y control de plataformas para el ocio”, VI Congreso Internacional de Diseño, Redes de Investigación y Tecnología para todos (DRT4ALL), Madrid, 2015 pp. 667-692.
- [15] M. Vidal, K. Pfeuffer, A. Bulling, H. Gellersen,” Pursuits: Eye-Based Interaction with Moving Targets”, The ACM SIGCHI Conference on Human Factors in Computing Systems, Paris, 2013, pp. 3147-3050.
- [16] H. Liang, J. Chang, I. K. Kazmi, J.J. Zhang, P. Jiao,” Hand gesture-based interactive puppetry system to assist storytelling for children”, in The Visual Computer, vol 33, pp 517–531, Apr. 2017.
- [17] R. McAdam, K. Nesbitt, “Leveraging Human Movement in the Ultimate Display”, Proceedings of the Thirteenth Australasian User Interface Conference (AUIC2013), Melbourne, 2013, pp. 11-20.
- [18] E. Hayashi, M. Maas, Jason I. Hong,” Wave to Me: User Identification Using Body Lengths and Natural Gestures”, CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14), Toronto,2014, pp. 3453-3462.
- [19] M. Heijboer, E. van den Hoven, B. Bongers, S. Bakker,” Facilitating peripheral interaction: design and evaluation of peripheral interaction for a gesture-based lighting control with multimodal feedback”, in Personal and Ubiquitous Computing, vol. 20, no 1, pp 1–22, Dec. 2015.
- [20] B. De Carolis, Giuseppe Palestra, “Gaze-based Interaction with a Shop Window”, International Working Conference on Advanced Visual Interfaces, Bari, 2016, pp. 304-305.
- [21] A. Mehler, A Lücking, G. Abrami, “WikiNect: image schemata as a basis of gestural writing for kinetic museum wikis” in Universal Access in the Information Society, vol. 14, no. 3, pp 333–349, Sep 2015.
- [22] M. Barrón-Estrada, R. Zatarain-Cabada, C. Aispuro-Gallegos, C. de la Luz Sosa-Ochoa, M. Lindor Valdez, “Java Tutoring System with Facial and Text Emotion Recognition”. International Journal of Advanced Research in Computer Science. 106. 49. 2015.

- [23] C. Wu, A. Quigley, D. Harris-Birtill, “Out of sight: a toolkit for tracking occluded human joint positions” in *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp 125–135, Feb. 2017.
- [24] C. W. Kwan, I. Paquette, J. J. Magee, M. Betke, “Adaptive sliding menubars make existing software more accessible to people with severe motion impairments” in *Universal Access in the Information Society*, Vol 13, no.1, pp 5–22, Mar. 2014.
- [25] R. J.M.G. Tello, C. Torturella, T. F. Bastos-Filho, “Control de una Silla de Ruedas Robótica de Alto Rendimiento por Medio de Potenciales Evocados Visuales”, VI Congreso Internacional de Diseño, Redes de Investigación y Tecnología para todos (DRT4ALL), Madrid, 2015 pp. 369-390.
- [26] A. L. Cardoso Bissoli, Y. Luduvico Coelho, T. Freire Bastos-Filho, “A System for Multimodal Assistive Domotics and Augmentative and Alternative Communication”, IX Conferencia Internacional de ACM sobre Tecnologías Difíciles Relacionadas con Ambientes Asistenciales, de Corfú, 2016, pp. 1-8.
- [27] R.N. Taylor, N. Mendvidovic, E. M. Dashofy, “Software Architecture” *Foundations, Theory and Practice*, John Wiley & Sons, 2009, pp. 94-97.
- [28] “Plano cartesiano” [online]. Available: http://www.ppsk12.or.us/district/depts/edmedia/videoteca/prope/htmlb/SEC_39.H M.